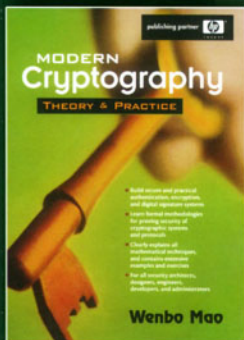




# 现代密码学 理论与实践

Modern Cryptography:  
Theory and Practice



[英] Wenbo Mao 著

王继林 伍前红 等译

王育民 姜正涛 审校



电子工业出版社

Publishing House of Electronics Industry

<http://www.phei.com.cn>

# 前 言

我们的社会已经进入了一个崭新时代,传统的商务活动、事务处理以及政府服务已经或越来越多地将要通过开放的计算机和通信网,如 Internet,特别是基于万维网的工具来实施和提供。对在世界各个角落的人来说,在线工作有着“随时可得”的巨大优点。下面是一些可以或即将可以在线完成的事例:

银行业务、账单支付、家中购物、股票交易、拍卖、税收、赌博、小额支付(例如按下载支付)、电子身份、对医药记录的在线访问、虚拟保密网、安全数据存档与恢复、文件的挂号递送、敏感文件的公平交换、公平合同签署、时戳、公正、选举、广告、授权、订票、交互式游戏、数字图书馆、数字权限管理、盗版追踪等。

只有在开放网络能提供安全通信的条件下,上述诱人的商务活动、事务处理以及服务才能实现。而要保证在开放网络中通信的安全性,一个有效的解决办法就是利用密码技术。加密、数字签名、基于口令的用户认证是实现安全通信的一些最基本的密码技术。但是,正如我们将在本书中多次看到的那样,即使是最基本的密码技术,在应用中也存在令人惊讶的难以捉摸和严重的安全性问题。而且对很多像上一段所列出的“想像的”应用来说,这些基本的密码技术是不够的。

越来越复杂的电子商务、事务处理和服务形式<sup>①</sup>,对在开放的网络中实现安全通信的需求正迅速增加。对能够进行设计、开发、分析和维护信息安全系统和密码协议的信息安全方面的专业人员的需求量正日益增大。这些专业人员可能是从 IT 系统的管理员、信息安全工程师和有安全要求的软/硬件系统产品的开发人员,直到密码学家。

在过去的几年里,作者作为在英国布里斯托尔 Hewlett-Packard 实验室信息安全与密码系统方面的一名技术顾问,已经注意到了对信息安全人员需求的持续增长和现有专业人员明显短缺这一失调现象。结果,很多在密码或信息安全方面缺少适当训练的面向应用的工程师,由于应用的需要,不得不“挽起袖子”成了安全系统或密码协议的设计者或者开发者。尽管这是不争的事实,但要设计密码系统和协议,即便对密码学专家来说,也不是件容易的事。

作者的工作性质允许他有机会审查很多信息安全系统和密码协议,其中有一些就是由“挽袖子”工程师所提出和设计的,而且是用在一些重要的实用上。在很多场合,作者看到了这些系统中存在有所谓的“教科书式密码”的特征,这是把很多密码学教科书中一般都介绍的密码算法直接拿来应用的结果。利用基本的公钥加密算法(如 RSA)直接对口令(一个不大的秘密数)进行加密就是“教科书式密码”的一个典型例子。教科书式密码以“不可忽略的概率”在重要应用场合下的出现引起了作者的担心。看来,教科书式密码的一般危害,尚没有被那些针对重要现实应用来设计和开发信息安全系统的许多人所意识到。

---

<sup>①</sup> Gartner Group 预计,欧盟的 B2B 和 B2C 电子商务税收在 2004 年将以 0.7 的概率达到 2.6 万亿美元,是 2000 年的 28 倍[5]。eMarketer[105]也报告,美国金融机构在 2002 年因电子身份问题被窃的损失为 140 亿美元,并预计每年将会以 29% 的速度增加。

考虑到对信息安全专业人才的大量需求,并相信专业人才的密码学知识不能仅囿于教科书式密码学,作者写了这本“非教科书式密码学”教材。本书致力于:

- 在强调“非教科书式”的情况下,广泛介绍有关密码算法、方案和协议。
- 通过展示对这类系统的大量攻击和总结典型的攻击技术,来说明“教科书式密码”的不安全性。
- 通过对标准的关注,为密码系统和协议的设计、分析与实施提供原理和指导原则。
- 研究严格建立密码系统和协议的强而实用安全性表示的形式化技术和方法。
- 为希望系统了解这一领域的读者精心选取学习现代密码学必备的理论素材。

## 本书范围

在过去的 30 年里,现代密码学的研究可谓突飞猛进,其研究领域非常广泛和深入。本书集中讨论一个方面的问题:对以其强安全性能明确建立起来的实用密码方案和协议进行介绍。

本书分为 6 部分:

**第一部分** 这一部分共有两章内容(第 1 章,第 2 章),是本书和密码学与信息安全入门性介绍。其中第 1 章以解决一个微妙的通信问题为开场白,来阐述密码学的效用。本章将给出一个在电话上实现公平掷币的简单密码协议(本书的第一个协议)并对其进行讨论。然后对要研究领域的文化和“贸易”进行介绍。第 2 章使用一系列的简单认证协议,来表明该领域的一个不幸的事实:缺陷处处存在。

作为入门性介绍,这一部分是为想进入该领域的新手写的。

**第二部分** 这一部分介绍学习本书必备的数学背景知识,它包含 4 章内容(第 3 章~第 6 章)。只想“知其然”,即了解如何使用实用密码方案和协议的读者,可以跳过这一部分而基本上不影响大多数后续章节的阅读。要想知道“所以然”,即为什么这些方案和协议具有强安全性的读者将会发现,这里给出的数学背景知识是足够的。当我们揭示方案和工作原理,指出其中的一些方案和协议是不安全的,或者论述别的协议和方案是安全的时候,我们就能在这里找到相应的理论根据。

这一部分也可作为学习现代密码学理论基础的系统背景知识。

**第三部分** 这部分也有 4 章内容(第 7 章~第 10 章),介绍提供保密和数据完整性保护最基本的密码算法和技术。其中第 7 章是对称加密方案,第 8 章是非对称加密技术,第 9 章讨论的是,在数据是随机的理想条件下,利用基本通用的非对称密码函数所拥有的一个重要的安全特性。最后的第 10 章是数据完整性技术。

由于这里介绍的是最基本的方案和技术,其中多数属于“教科书式密码”,因而不安全的。在介绍这些方案的同时,也给出不少相应的攻击,并明确阐述了告诫注释。对于那些不想对实用密码和它们的强安全性概念做深入研究的实际工作人员,教科书式密码部分也仍会就教科书式密码的不安全性向他们提出明确的预警信号。

**第四部分** 这部分有 3 章内容(第 11 章~第 13 章),介绍应用密码学和信息安全中一个重要的概念——认证。这些章节的研究范围很广,第 11 章介绍技术背景、原理、一系列的基础协议和标准,以及常规的攻击技术和防护措施。第 12 章是对四个著名的认证协议系统在现实应用案例的研究。第 13 章介绍特别适合于开放系统的有关最新技术。

企业中信息安全系统的管理者、安全产品的软/硬件开发商们将会发现这一部分对他们是非常有用的。

**第五部分** 这部分有 4 章内容(第 14 章~第 17 章),对公钥密码技术(加密、签名和签密)的强(实用)安全性概念进行严格的形式化处理,并给出认证协议的形式化分析方法。第 14 章介绍强安全性概念的形式化定义,接着的两章是和第三部分教科书式密码方案相对的实用密码方案,具有形式化建立起(即明确推理)的强安全性。最后,第 17 章对在第四部分尚未进行分析的认证协议,给出其形式化的分析方法和技术。

**第六部分** 这是本书的最后一部分,包括两个技术章节(第 18 章~第 19 章)和一个简短的评述(第 20 章)。其主要的技术章节,第 18 章,介绍了一类被称为零知识协议的密码协议。这类协议能提供一种重要的安全性业务,它为“想像中”的各种电子商务和事务处理应用所必需,在对所声明的内容保持严格保密性的情况下,对一个秘密数所宣称的性质进行证实(例如,符合商业上的需求)。这部分要引入零知识协议,说明了在各种现实应用中对特定安全性需求的多样性。这种多样性超越了机密性、完整性、认证和不可否认性。在本书的最后一个技术章节(第 19 章)中,我们将解决本书一开始介绍的协议中的遗留问题——实现“通过电话公平掷币”。最后的实现协议不但在效率上适宜实用,而且也明确地建立起了强安全性。

不用说,对每一个实用密码协议或方案的描述,都是要先给出与之相应的教科书式密码不适用的原因。我们总是以给出相应存在的攻击来说明原因,就相应的攻击而言,这些方案和协议往往存在某些微妙之处。另外,一个实用密码协议和方案的描述,也必是以其所宣称的必须包含的强(实用)安全性成立的分析来结束。因而,本书一些部分不可避免地要包含有数学和逻辑推理、为明示和对付攻击所需的归纳和变换。

实用密码学并不是一个轻易驾驭或者稍稍读读就能掌握的论题。尽管如此,本书并不是一本仅为专业密码学家所感兴趣的高深研究课题的书。这里所介绍的东西对密码学家来说都是已知的和相当基本的。作者相信,在有充足的解释、实例、足够的数学背景知识和参考材料的情况下,这些东西完全能被非专业人士理解。

本书针对的读者对象为:

- 已经或即将完成计算机、信息科学、应用数学第一学位课程并计划从事信息安全行业的学生。对他们来说,本书可以作为应用密码学的高级教程。
- 在高科技公司从事信息安全系统设计和开发的安全工程师。如果我们说在科学研究计划中,教科书式密码所产生的危害不是很大的话,至多是出现一种令人尴尬的场面,那么在信息安全产品中使用教科书式密码将会造成严重损失。因此对这类读者来说,了解教科书式密码在现实中的不适用性是非常必要的。而且,这类读者应该对隐藏在实用方案和协议下的安全原理有很好的理解,以便能正确地使用这些方案和原理。第 II 部分所给出的自足的数学基础材料使得本书很适合这类读者自学。
- 对企业信息安全系统管理人员或者生产安全产品的软/硬件开发商这类读者来说,第 I 部分是简单而基本的文化和“商务”方面的培训教程,第 III 部分和第 IV 部分是一个适当裁减的密码学和信息安全知识集。这三部分包含有很多基本的密码方案和协议,以



及很多对应的攻击方法和防护措施。这些攻击方法和防护措施能被大多数读者理解,不需要有所谓理论基础的负担。

- 对于刚开始从事密码学或计算机安全方面研究的博士生这类读者来说,将会欣赏一本能包含强安全性概念的形式化处理并对其进行适当和详细解释的书感兴趣。本书将能帮助他们快速深入地进入这一浩瀚的研究领域。对这类人员来说,第 II、IV、V 和 VI 部分构成了一个适当深度的文献综述材料,这将能引导他们找到更进一步的文献,并能帮助他们明确自己的研究课题。
- 本书的适当取舍(如第 1 章、第 2 章、第 3 章和第 6 章)也可以组成适合计算机、信息科学和应用数学大学高年级学生学习用的应用密码学教程。

## 致谢

非常感谢 Feng Bao、Colin Boyd、Steven Galbraith、Dieter Gollmann、Keith Harrison、Marcus Leech、Helger Lipmaa、Hoi-Kwong Lo、Javier Lopez、John Malone-lee、Cary Meltzer、Christian Paquin、Kenny Paterson、David Pointcheval、Vincent Rijmen、Nigel Smart、David Soldera、Paul van Oorschot、Serge Vaudenay 和 Stefek Zaba。他们花费了大量时间校阅有关章节或全书,并提供了非常有价值的评论、批评和建议,使得本书更加完善。

本书还得益于向下列人士的请教: Mihir Bellare、Jan Camenisch、Neil Dunbar、Yair Frankel、Shai Halevi、Antoine Joux、Marc Joye、Charlie Kaufman、Adrian Kent、Hugo Krawczyk、Catherine Meadows、Bill Munro、Phong Nguyen、Radia Perlman、Marco Ricca、Ronald Rivest、Steve Schneider、Victor Shoup、Igor Shparlinski 和 Moti Yung。

作者还要感谢 Prentice-Hall PTR 的 Jill Harry 和 HP Professional Books 的 Susan Wright,他们鼓励并引导我写作了本书,并在我漫长的写作中提供了技术帮助,感谢 Prentice-Hall PTR 的 Jennifer Blackwell、Robin Carroll、Brenda Mulligan、Justin Somma 和 Mary Sudul 以及 HP Professional Books 的 Walter Bruce 和 Pat Pekary。

还要感谢我在布里斯托尔 Hewlett-packard 实验室的同事们在技术、修辞和管理方面给予的支持,他们是 David Ball、Rachard Cardwell、Liqun Chen、Lan Cole、Gareth Jones、Stephen Pearson 和 Martin Sadler。

作者于英国布里斯托尔

2003 年 5 月

# 目 录

第一部分 引言 .....	1
第 1 章 一个简单的通信游戏 .....	2
1.1 一个通信游戏 .....	2
1.1.1 我们给出密码学的第一个应用示例 .....	2
1.1.2 对密码学基础的初步提示 .....	4
1.1.3 信息安全基础:计算困难性的背后 .....	4
1.1.4 密码学的新作用:保证游戏的公平性 .....	5
1.2 描述密码系统和协议的准则 .....	6
1.2.1 保护的程度与应用需求相符合 .....	6
1.2.2 对安全性的信心要依据所建立的“种系” .....	7
1.2.3 实际效率 .....	8
1.2.4 采用实际的和可用的原型和服务 .....	8
1.2.5 明确性 .....	9
1.2.6 开放性 .....	12
1.3 本章小结 .....	12
习题 .....	13
第 2 章 防守与攻击 .....	14
2.1 引言 .....	14
2.1.1 本章概述 .....	14
2.2 加密 .....	14
2.3 易受攻击的环境(Dolev-Yao 威胁模型) .....	16
2.4 认证服务器 .....	17
2.5 认证密钥建立的安全特性 .....	18
2.6 利用加密的认证密钥建立协议 .....	19
2.6.1 消息保密协议 .....	19
2.6.2 攻击、修复、攻击、修复 .....	21
2.6.3 消息认证协议 .....	23
2.6.4 询问-应答协议 .....	25
2.6.5 实体认证协议 .....	27
2.6.6 一个使用公钥密码体制的协议 .....	28
2.7 本章小结 .....	31
习题 .....	32
第二部分 数学基础 .....	33
标准符号 .....	34
第 3 章 概率论和信息论 .....	36
3.1 引言 .....	36

3.1.1 本章纲要 .....	36
3.2 概率论的基本概念 .....	36
3.3 性质 .....	37
3.4 基本运算 .....	38
3.4.1 加法规则 .....	38
3.4.2 乘法规则 .....	38
3.4.3 全概率定律 .....	39
3.5 随机变量及其概率分布 .....	40
3.5.1 均匀分布 .....	40
3.5.2 二项式分布 .....	41
3.5.3 大数定律 .....	44
3.6 生日悖论 .....	45
3.6.1 生日悖论的应用:指数计算的 Pollard 袋鼠算法 .....	46
3.7 信息论 .....	48
3.7.1 熵的性质 .....	49
3.8 自然语言的冗余度 .....	50
3.9 本章小结 .....	51
习题 .....	51
<b>第 4 章 计算复杂性</b> .....	<b>53</b>
4.1 引言 .....	53
4.1.1 本章概述 .....	53
4.2 图灵机 .....	54
4.3 确定性多项式时间 .....	54
4.3.1 多项式时间计算性问题 .....	56
4.3.2 算法与计算复杂度表示 .....	57
4.4 概率多项式时间 .....	65
4.4.1 差错概率的特征 .....	66
4.4.2 “总是快速且正确的”子类 .....	68
4.4.3 “总是快速且很可能正确的”子类 .....	69
4.4.4 “很可能快且总是正确的”子类 .....	70
4.4.5 “很可能快且很可能正确的”子类 .....	72
4.4.6 有效算法 .....	77
4.5 非确定多项式时间 .....	78
4.5.1 非确定多项式时间完全 .....	81
4.6 非多项式界 .....	82
4.7 多项式时间不可区分性 .....	84
4.8 计算复杂性理论与现代密码学 .....	85
4.8.1 必要条件 .....	85
4.8.2 非充分条件 .....	86
4.9 本章小结 .....	87
习题 .....	87

<b>第5章 代数学基础</b>	89
5.1 引言	89
5.1.1 章节纲要	89
5.2 群	89
5.2.1 拉格朗日定理	91
5.2.2 群元素的阶	93
5.2.3 循环群	94
5.2.4 乘法群 $\mathbb{Z}_n^*$	96
5.3 环和域	97
5.4 有限域的结构	98
5.4.1 含有素数个元素的有限域	99
5.4.2 模不可约多项式的有限域	100
5.4.3 用多项式基构造有限域	104
5.4.4 本原根	107
5.5 用椭圆曲线上的点构造群	108
5.5.1 群运算	109
5.5.2 点乘	112
5.5.3 椭圆曲线离散对数问题	112
5.6 本章小结	113
习题	114
<b>第6章 数论</b>	115
6.1 引言	115
6.1.1 本章概述	115
6.2 同余和剩余类	115
6.2.1 $\mathbb{Z}_n$ 中运算的同余性质	116
6.2.2 求解 $\mathbb{Z}_n$ 中的线性同余式	117
6.2.3 中国剩余定理	118
6.3 欧拉 $\phi$ 函数	122
6.4 费马定理、欧拉定理、拉格朗日定理	123
6.5 二次剩余	124
6.5.1 二次剩余的判定	125
6.5.2 勒让德-雅可比符号	126
6.6 模一个整数的平方根	128
6.6.1 求模为素数时的平方根	128
6.6.2 求模为合数时的平方根	131
6.7 Blum 整数	133
6.8 本章小结	134
习题	134
<b>第三部分 基本的密码学技术</b>	137
<b>第7章 加密——对称技术</b>	138
7.1 引言	138

7.1.1 本章概述 .....	138
7.2 定义 .....	139
7.3 代换密码 .....	140
7.3.1 简单的代换密码 .....	140
7.3.2 多表密码 .....	142
7.3.3 弗纳姆密码和一次一密 .....	142
7.4 换位密码 .....	143
7.5 古典密码:使用 and 安全性 .....	144
7.5.1 古典密码的使用 .....	145
7.5.2 古典密码的安全性 .....	145
7.6 数据加密标准(DES) .....	146
7.6.1 介绍 DES .....	146
7.6.2 DES 的核心作用:消息的随机非线性分布 .....	148
7.6.3 DES 的安全性 .....	149
7.7 高级加密标准(AES) .....	150
7.7.1 Rijndael 密码概述 .....	150
7.7.2 Rijndael 密码的内部函数 .....	151
7.7.3 Rijndael 内部函数的功能小结 .....	154
7.7.4 快速而安全的实现 .....	154
7.7.5 AES 对应用密码学的积极影响 .....	155
7.8 运行的保密模式 .....	155
7.8.1 电码本模式(ECB) .....	156
7.8.2 密码分组链接模式(CBC) .....	157
7.8.3 密码反馈模式(CFB) .....	160
7.8.4 输出反馈模式(OFB) .....	160
7.8.5 计数器模式(CTR) .....	161
7.9 对称密码体制的密钥信道建立 .....	161
7.10 本章小结 .....	163
习题 .....	163
<b>第 8 章 加密——非对称技术</b> .....	<b>165</b>
8.1 引言 .....	165
8.1.1 本章概述 .....	166
8.2 “教科书式加密算法”的不安全性 .....	166
8.3 Diffie-Hellman 密钥交换协议 .....	167
8.3.1 中间人攻击 .....	168
8.4 Diffie-Hellman 问题和离散对数问题 .....	169
8.4.1 任意参数对于满足困难假设的重要性 .....	172
8.5 RSA 密码体制(教科书式) .....	173
8.6 公钥密码体制的分析 .....	175
8.7 RSA 问题 .....	176
8.8 整数分解问题 .....	177
8.9 教科书式 RSA 加密的不安全性 .....	179



8.9.1	中间相遇攻击和教科书式 RSA 上的主动攻击	179
8.10	Rabin 加密体制(教科书式)	181
8.11	教科书式 Rabin 加密的不安全性	182
8.12	ElGamal 密码体制(教科书式)	184
8.13	教科书式 ElGamal 加密的不安全性	186
8.13.1	教科书式 ElGamal 加密的中间相遇攻击和主动攻击	187
8.14	公钥密码系统需要更强的安全定义	187
8.15	非对称密码与对称密码的组合	188
8.16	公钥密码系统密钥信道的建立	189
8.17	本章小结	190
	习题	190
<b>第 9 章</b>	<b>理想情况下基本公钥密码函数的比特安全性</b>	<b>192</b>
9.1	前言	192
9.1.1	本章概述	192
9.2	RSA 比特	192
9.3	Rabin 比特	196
9.3.1	Blum-Blum-Shub 伪随机比特生成器	196
9.4	ElGamal 比特	196
9.5	离散对数比特	197
9.6	本章小结	199
	习题	199
<b>第 10 章</b>	<b>数据完整性技术</b>	<b>201</b>
10.1	引言	201
10.1.1	本章概述	201
10.2	定义	201
10.3	对称技术	202
10.3.1	密码杂凑函数	203
10.3.2	基于密钥杂凑函数的 MAC	205
10.3.3	基于分组加密算法的 MAC	206
10.4	非对称技术 I: 数字签名	206
10.4.1	数字签名的教科书式安全概念	208
10.4.2	RSA 签身体制(教科书式版本)	209
10.4.3	RSA 签字安全性的非形式化论证	209
10.4.4	Rabin 签名体制(教科书式版本)	210
10.4.5	关于 Rabin 签名的一个自相矛盾的安全性基础	210
10.4.6	ElGamal 签名体制	212
10.4.7	ElGamal 签名体制安全性的非形式化论证	212
10.4.8	ElGamal 签名族中的签名体制	215
10.4.9	数字签名体制安全性的形式化证明	218
10.5	非对称技术 II: 无源识别的数据完整性	218
10.6	本章小结	221
	习题	221

<b>第四部分 认证</b>	223
<b>第 11 章 认证协议——原理篇</b>	224
11.1 引言	224
11.1.1 章节概述	225
11.2 认证和细化的概念	225
11.2.1 数据源认证	225
11.2.2 实体认证	226
11.2.3 认证的密钥建立	227
11.2.4 对认证协议的攻击	227
11.3 约定	228
11.4 基本认证技术	229
11.4.1 消息新鲜性和主体活性	229
11.4.2 双方认证	235
11.4.3 包含可信第三方的认证	236
11.5 基于口令的认证	238
11.5.1 Needham 口令认证协议及其在 UNIX 操作系统中的实现	239
11.5.2 一次性口令机制(及缺陷的修补)	240
11.5.3 加盐操作:加密的密钥交换(EKE)	242
11.6 基于非对称密码学的认证密钥交换	244
11.6.1 工作站-工作站协议	245
11.6.2 简化 STS 协议的一个缺陷	246
11.6.3 STS 协议的一个瑕疵	248
11.7 对认证协议的典型攻击	250
11.7.1 消息重放攻击	250
11.7.2 中间人攻击	251
11.7.3 平行会话攻击	252
11.7.4 反射攻击	253
11.7.5 交错攻击	255
11.7.6 归因于类型缺陷攻击	255
11.7.7 归因于姓名遗漏攻击	256
11.7.8 密码服务滥用攻击	257
11.8 文献简记	260
11.9 本章小结	260
习题	261
<b>第 12 章 认证协议——实践篇</b>	262
12.1 引言	262
12.1.1 章节概述	263
12.2 用于因特网的认证协议	263
12.2.1 IP 层通信	263
12.2.2 IP 安全协议(IPSec)	264
12.2.3 因特网密钥交换(IKE)协议	267
12.2.4 IKE 中看似合理的可否认性	272

12.2.5	对 IPSec 和 IKE 的批评意见 .....	273
12.3	安全壳(SSH)远程登录协议 .....	274
12.3.1	SSH 架构 .....	274
12.3.2	SSH 传输层协议 .....	275
12.3.3	SSH 策略 .....	277
12.3.4	警告 .....	277
12.4	Kerberos 协议及其在 Windows 2000 系统中的实现 .....	278
12.4.1	单点登录结构 .....	279
12.4.2	Kerberos 交换 .....	280
12.4.3	警告 .....	282
12.5	SSL和TLS .....	282
12.5.1	TLS 架构概述 .....	283
12.5.2	TLS 握手协议 .....	283
12.5.3	TLS 握手协议的典型运行 .....	285
12.5.4	对 TLS 协议的边信道攻击 .....	286
12.6	本章小结 .....	287
习题	.....	288
<b>第 13 章</b>	<b>公钥密码的认证框架</b> .....	289
13.1	前言 .....	289
13.1.1	本章概述 .....	289
13.2	基于目录的认证框架 .....	289
13.2.1	证书发行 .....	291
13.2.2	证书吊销 .....	291
13.2.3	公钥认证框架实例 .....	291
13.2.4	与 X.509 公钥证书基础设施相关的协议 .....	293
13.3	基于非目录的公钥认证框架 .....	293
13.3.1	Shamir 的基于 ID 的签名方案 .....	294
13.3.2	基于 ID 的密码确切提供了什么 .....	295
13.3.3	自证实公钥 .....	296
13.3.4	利用“弱”椭圆曲线对构造基于身份的公钥密码体制 .....	298
13.3.5	Sakai、Ohgishi 和 Kasahara 的基于 ID 的非交互密钥分享系统 .....	301
13.3.6	三方 Diffie-Hellman 密钥协商 .....	303
13.3.7	Boneh 和 Franklin 的基于 ID 的密码体制 .....	304
13.3.8	非交互特性:无密钥信道的认证 .....	307
13.3.9	基于身份的公钥密码学的两个公开问题 .....	307
13.4	本章小结 .....	308
习题	.....	308
<b>第五部分</b>	<b>建立安全性的形式化方法</b> .....	311
<b>第 14 章</b>	<b>公钥密码体制的形式化强安全性定义</b> .....	312
14.1	引言 .....	312
14.1.1	本章概述 .....	313

14.2	安全性的形式化处理 .....	313
14.3	语义安全性——可证明安全性的首次亮相 .....	316
14.3.1	SRA 智力扑克协议 .....	316
14.3.2	基于教科书式安全的安全性分析 .....	317
14.3.3	Goldwasser 和 Micali 的概率加密 .....	319
14.3.4	GM 密码体制的安全性 .....	320
14.3.5	ElGamal 体制的一种语义安全版本 .....	321
14.3.6	基于 Rabin 比特的语义安全密码体制 .....	323
14.4	语义安全性的不充分性 .....	324
14.5	超越语义安全性 .....	326
14.5.1	抗击选择密文攻击的安全性 .....	326
14.5.2	抗击适应性选择密文攻击的安全性 .....	329
14.5.3	不可展密码学 .....	331
14.5.4	不可区分性与不可展性的关系 .....	333
14.6	本章小结 .....	336
	习题 .....	337
<b>第 15 章</b>	<b>可证明安全的有效公钥密码体制 .....</b>	<b>338</b>
15.1	引言 .....	338
15.1.1	本章概述 .....	338
15.2	最优非对称加密填充 .....	339
15.2.1	安全性证明的随机预言机模型 .....	340
15.2.2	RSA-OAEP .....	342
15.2.3	RSA-OAEP 证明中的曲折 .....	342
15.2.4	对 RSA-OAEP 的补救工作 .....	349
15.2.5	RSA-OAEP“归约为矛盾”的严谨性 .....	351
15.2.6	对随机预言机模型的批评 .....	352
15.2.7	作者对随机预言机模型价值的观点 .....	352
15.3	Cramer-Shoup 公钥密码体制 .....	353
15.3.1	在标准困难性假设下的可证明安全性 .....	353
15.3.2	Cramer-Shoup 体制 .....	354
15.3.3	安全性证明 .....	357
15.4	可证明安全的混合密码体制综述 .....	362
15.5	可证明安全的实用公钥密码体制的文献注记 .....	364
15.6	本章小结 .....	365
	习题 .....	366
<b>第 16 章</b>	<b>强可证明安全的数字签名方案 .....</b>	<b>367</b>
16.1	引言 .....	367
16.1.1	本章纲要 .....	368
16.2	数字签名的强安全性定义 .....	368
16.3	ElGamal 族签名的强可证明安全 .....	369
16.3.1	三元组 ElGamal 族签名 .....	369
16.3.2	分叉归约技术 .....	370

16.3.3	重行归约方法	376
16.4	适于应用的 RSA 和 Rabin 签名方法	377
16.4.1	具有随机化填充的签名	377
16.4.2	概率签名方案	377
16.4.3	PSS-R: 消息可恢复的签名	379
16.4.4	签名和加密通用的 PSS-R 填充	379
16.5	签密	381
16.5.1	Zheng 的签密方案	382
16.5.2	一箭双雕: 采用 RSA 签密	384
16.6	本章小结	387
习题		388
<b>第 17 章</b>	<b>分析认证协议的形式化方法</b>	389
17.1	引言	389
17.1.1	本章概述	390
17.2	认证协议的形式化描述	390
17.2.1	加解密认证方法的不精确性	390
17.2.2	认证协议的细化描述	393
17.2.3	认证协议细化描述的例子	394
17.3	正确协议的计算观点——Bellare-Rogaway 模型	397
17.3.1	参与者行为的形式模型化	398
17.3.2	相互认证的目标: 匹配对话	400
17.3.3	MAP1 协议及其安全性证明	401
17.3.4	协议正确性计算模型的进一步研究	402
17.3.5	讨论	403
17.4	正确协议的符号操作观点	403
17.4.1	定理证明	403
17.4.2	一种认证逻辑	404
17.5	形式化分析技术: 状态系统探查	406
17.5.1	模型检验	406
17.5.2	NRL 协议分析机	408
17.5.3	CSP 方法	409
17.6	调和安全性形式化技术的两种观点	413
17.7	本章小结	414
习题		414
<b>第六部分</b>	<b>密码学协议</b>	417
<b>第 18 章</b>	<b>零知识协议</b>	418
18.1	引言	418
18.1.1	本章纲要	419
18.2	基本定义	419
18.2.1	计算模型	419
18.2.2	交互式证明协议的形式化定义	419



18.2.3	一个复杂性理论结果	422
18.3	零知识特性	423
18.3.1	完备零知识	424
18.3.2	诚实验证者的零知识	427
18.3.3	计算零知识	429
18.3.4	统计零知识	431
18.4	证明还是论据	432
18.4.1	零知识论据	432
18.4.2	零知识证明	433
18.5	双边差错协议	434
18.5.1	零知识证明双素整数	435
18.6	轮效率	438
18.6.1	子群成员归属的轮效率下界	439
18.6.2	离散对数的常数轮证明	441
18.7	非交互式零知识	444
18.7.1	利用指定验证者获得 NIZK	445
18.8	本章小结	448
习题		448
<b>第 19 章</b>	<b>回到“电话掷币”协议</b>	<b>450</b>
19.1	Blum“电话掷币”协议	450
19.2	安全性分析	451
19.3	效率	452
19.4	本章小结	453
<b>第 20 章</b>	<b>结束语</b>	<b>454</b>
<b>参考文献</b>		<b>455</b>

# 第一部分 引言

本书第一部分由两个引论性的篇章组成。主要为我们介绍密码学和信息安全中的一些基本概念;我们进行通信与处理敏感信息的环境;在该环境下参与演出的几位著名“人物”以及其中一些扮演坏家伙的标准伎俩;密码学和信息安全系统研究与发展领域的社团文化,以及这些系统极为容易出错的现实。

作为初级引论,这一部分所针对的读者是希望进入这一领域的初学者。

# 第 1 章 一个简单的通信游戏

作为本书的开场白,我们用应用密码学中的一个简单例子来解决一个简单问题。这个例子对我们要在本书中介绍的内容有三个方面的作用:

- 初步揭示了应用密码技术解决应用中的敏感问题的效力和实用性。
- 对密码学的基础知识进行了初步提示。
- 初步建立了从事信息安全的密码系统的开发所要求的思考形式。

首先,我们将给出一个平凡且简单的问题,然后给出同样简单的相应解决办法。该解决办法是我们大家都熟悉的二人博弈。可是我们将会发现,当两个参与者彼此相距遥远时,这个简单游戏将马上变得麻烦起来。游戏双方彼此的物理分离就会丧失公平游戏的基础,从而会引出麻烦,玩游戏的各方不可能相信另一方在公平地进行游戏。

远距离参与者公平进行游戏的需求将“激励”我们,利用盔甲的屏障进行防护来强化我们的简单游戏。这种强化游戏的方法来自长期树立起来的、在公开网络中保护通信的思想,即利用密码技术隐藏信息。

在应用密码技术很好地解决了我们的第一个安全问题之后,我们将对密码系统的质量准则进行一系列的讨论(见 1.2 节),这些讨论是我们为保护敏感信息所进行的研究和开发的技术领域的背景知识和文化的介绍。

## 1.1 一个通信游戏

这是个简单问题。两个朋友, Alice 和 Bob<sup>①</sup>想在晚上一起外出,但是他们定不下来是去电影院还是歌剧院。尽管如此,他们达成了—个通过掷硬币来决定的协议,这和我们都很熟悉的掷硬币游戏—样。

Alice 拿着一个硬币并对 Bob 说:“你选择一面,然后我来抛”。Bob 选择后, Alice 把硬币抛向空中。然后他们都注视硬币,看结果是哪一面朝上。如果 Bob 选择的那面朝上,则他就可决定要去的地方,否则由 Alice 决定。

在通信规程的研究中,对上述的一个多方游戏可以给予一个更“科学的”名字:协议。一个协议是一个适当定义的、在多个参与实体之间执行的规程。这里要注意多个实体参与的重要性:如果一个规程仅由一个实体执行,那么它只是一种程序,不能称之为协议。

### 1.1.1 我们给出密码学的第一个应用示例

现在假想这两个朋友尝试在电话上执行上述协议, Alice 向 Bob 说:“你选一面,然后我抛硬币并告诉你是否赢了”。显然 Bob 不会同意,因为他不能验证抛掷硬币的结果。

---

<sup>①</sup> 在密码学、密码协议和信息安全领域, Alice 和 Bob 都是著名的角色,他们将在本书的大多数协议中出现。

然而,我们可以在这个协议中加入一点密码技术,把它变成一个适合在电话上工作的形式。其结果就成为一个密码协议,即本书的第一个密码协议!现在先让我们把所用到的“密码术”看做是一个数学函数  $f(x)$ ,它是整数上的映射并具有下列奇妙性质:

### 性质 1.1 奇妙函数 $f$

- I) 对任意整数  $x$ ,由  $x$  计算  $f(x)$  是容易的,而给出  $f(x)$ ,要找出对应的原像  $x$  是不可能的,不管  $x$  是奇数还是偶数。
- II) 不可能找出一对整数  $(x, y)$ ,满足  $x \neq y$  且  $f(x) = f(y)$ 。

在性质 1.1 中,形容词“容易”和“不可能”的意思需要做进一步的阐述。同样,由于这些词与困难的程度有密切关系,我们必须搞清楚它们的量化表示。可是,因为现在我们把  $f(x)$  看成是一个特殊类型的函数,采用常规语言使用这些词依然是安全的。在第 4 章我们将为本书中使用的各种“容易”和“不可能”提供数学公式描述。本书的重要任务之一就是为各种“容易”、“困难”和“不可能”建立量化表示。事实上,我们最终将在本书最后的技术性篇章(第 19 章)看到,在我们最终所实现的掷币协议中,性质 1.1 中奇妙函数  $f$  “不可能”的两种用法将有大不相同的定量量度。

假定两个朋友已经就奇妙函数  $f(x)$  达成了一致,并一致同意用偶数来表示“正面”,用奇数来表示“背面”。现在他们做好了在电话上执行我们第一个密码协议(协议 1.1)的准备。

---

### 协议 1.1 电话掷币

假定

Alice 和 Bob 已经同意:

- i) 具有性质 1.1 的一个特殊函数  $f$
- ii)  $f(x)$  中的偶数  $x$  代表“正面”,奇数  $x$  代表“背面”

(\* 注意:由于(ii),该协议有一个弱点,详见习题 1.2 \*)

1. Alice 选择一个大随机数  $x$  并计算  $f(x)$ ;然后通过电话告诉 Bob  $f(x)$  的值;
2. Bob 告诉 Alice 自己对  $x$  的奇偶性猜测;
3. Alice 告诉 Bob  $x$  的值;
4. Bob 验证  $f(x)$  并察看他所做的猜测是正确或错误。

---

要说明上述“电话掷币协议”能在电话上很好地工作并不困难,下面是对其初步的“安全性分析”(注意,这里为安全性分析加上引号是因为我们在这里所给出的分析是远远不够的)。

#### 1.1.1.1 初步的“安全性分析”

首先,根据  $f$  具有的“性质 II”,Alice 无法找到不同的两个数  $x$  和  $y$ ,其中一个是奇数而另一个是偶数(可表示为  $x \neq y \bmod 2$ ),使其满足  $f(x) = f(y)$ 。因此,Alice 一旦通过电话告诉 Bob  $f(x)$  的值(第 1 步),她也就向 Bob 就  $x$  的值做出了承诺,她无法再改变  $x$  的值。也就是说 Alice 已经完成了其掷硬币过程。

第二,由于  $f$  具有“性质 I”,已知  $f(x)$ ,Bob 不能判定出 Alice 所使用的  $x$  是奇数还是偶数,因而他不得不把其猜测(第 2 步)真实地给出(即,并非通过各种推导得到的)。这样,Alice 可

给出  $x$  的值令 Bob 相信其猜测是否正确(第 3 步)。事实上,如果 Bob 利用 Alice 告诉的  $x$  对  $f(x)$  进行计算的结果(第 4 步)与 Alice 在第 1 步给出的结果一样,且 Bob 相信  $f$  所具有的性质,则 Bob 应该相信最终的输赢。而且,在  $x$  选自一个充分大的空间且 Bob 的猜测无优(即获胜的概率不大于  $1/2$ )的情况下,掷硬币是公平的。

应该指出,在对协议 1.1 的“安全性分析”中,我们已做了若干简化和省略。这样,当前的协议版本还远不能用于具体的实际。本章将讨论其中的一些简化和省略。不过,正确而具体地实现该协议所必要的技术,以及分析其安全性所必要的方法,将是本书后续章节要讨论的主题。我们将把对协议 1.1 正确而具体的实现(更确切地说是“奇妙函数” $f$ )推迟到本书最后一章(第 19 章)。在那里,我们已做好了技术准备,将能对其具体实现提供一个形式化的安全性分析。

### 1.1.2 对密码学基础的初步提示

虽然上述第一个协议很简单,但它的确是一个合格的密码协议,因为协议中使用的“奇妙函数”是构成现代密码学的一个基本要素——单向函数。性质 1.1 所列出的两个奇妙特性提出了两个计算困难性问题,其中一个对应 Alice,另一个对应 Bob。

根据对协议 1.1 的初步安全性分析,我们可以断言,单向函数的存在性意味着在娱乐场地进行安全选取的可能性。下面是这一论断的合理推广:

单向函数的存在性意味着安全密码系统的存在性。

现在也已众所周知,上述断言的逆命题也是成立的:

安全的密码系统的存在性意味着单向函数的存在性。

目前普遍认为确实存在有单向函数。因而在保护信息方面我们是乐观的。我们的乐观态度常常能被我们的日常经验所证实:现实中的很多过程,不管是数学的或是其他方面的,都具有单向性。考虑下面的物理现象(虽然不是一个非常精确的数学类比):茶杯掉在地上摔成碎片并耗散一定的能量进入周围环境中(例如,热、声音或甚至一点暗淡的光)是一个容易的过程。而相反的过程,即重新将耗散的能量收集起来,并用来把碎片整合成一个完整的杯子,即便不是不可能的,也必然是很困难的(如果可能,则整个收集起来的能量就能够把重新整合的杯子弹回到其开始降落时的高度)!

在第 4 章,我们将看到一类数学函数,这类函数为现代密码学提供所需的单向性质。

### 1.1.3 信息安全基础:计算困难性的背后

我们已经断言,信息安全性需要某些数学特性。而且,我们还做了一个乐观的论断:数学特性意味着(即保证了)信息安全性。

但在现实中,后面的断言并不无条件成立!现实应用的安全性依赖于很多问题。让我们继续以第一个协议例子来给予说明。

应该指出,在对协议 1.1 的初步安全性分析中,有很多重要问题我们没有考虑。事实上,协议 1.1 本身是一个非常简化的版本,它省略了一些细节,这些细节对于设计的协议所要提供的安全性服务是非常重要的。这些省略阻碍了我们询问一些问题。

例如,我们可以怀疑:Alice 真的“被迫”不改变她所使用的  $x$ ? 同样,Bob 真的“被迫”不改变他对  $x$  的奇偶性猜测? 这里“被迫”的意思是指,不管电话上的声音是否足以保证上述强数



学性质起作用。我们还可以问, Alice 是否有一个好的随机数发生器使她能获得随机数  $x$ 。这个问题在很多需要做出公平判定的更为关键的应用中是至关重要的。

所有这些细节都在该简化协议的说明中省略了, 因而它们都变成了隐含的假定(后面有更详细的叙述)。事实上, 如果用这个协议做更严格的判定, 协议还应该包含一些明确的用法说明。例如, 当读出  $f(x)$  的值和  $x$  的奇偶性猜测时, 两个参与者都要考虑记录对方的电话录音, 以便在发生争议时使用。

一些密码系统和协议, 特别是密码教科书所介绍的, 与“电话掷币”协议类似, 常以简化的形式给出。采用简化的形式有助于更清楚地陈述, 特别当一些约定被认为是显然的时候。但是有时候, 隐含的约定或假设可能是非常微妙的, 并可能会带来意想不到的后果。这对于本想通过省略一些细节达到更“清楚陈述”的做法多少有些讽刺意味。安全系统中一个假定的不成立, 可能会招致某种形式的攻击, 并因而导致所提供的服务成为泡影。要意识到一个隐含的假定不成立是相当困难的。在 1.2.5 节中, 我们将对有关密码系统的明确设计和规范的重要性进行讨论。

本书的一个主要论题就是要说明在现实应用的安全性中, 有很多与应用相关的微妙之处需要认真对待。

#### 1.1.4 密码学的新作用: 保证游戏的公平性

密码学一度曾为政府所独占。军事和外交部门使用它来保密消息。可是今天, 密码学除了用于对信息进行保密外, 还有一个新的用途: 在有大量“玩家”的“游戏”中保证公平性。这也是我们选用一个通信游戏作为论述密码学的本书开场的部分原因。

在一个娱乐场所进行判决可能不是一件大不了的事情, 因而通过电话掷硬币来做决定只可能被看成是一种取乐的通信游戏。可是, 有很多通信“游戏”必须更加认真地对待。随着越来越多的事务处理和电子商务活动在开放的网络中以电子方式开展, 我们通信中的许多实例将会涉及各种各样的“游戏”(在本书的前言中, 我们已经列出了在开放网络上开展的很多事务处理和服务的例子, 所有这些例子都包含参与者按一套规则的交互式活动, 都可以看成是“玩通信游戏”)。这些“游戏”可能是非常重要的!

一般来说, 这种“游戏”的“玩家”往往彼此物理上相距很远, 要依靠不安全的开放网络进行通信。物理距离和缺少安全性组合到一块儿, 有助于和/或激励一些“玩家”(甚至一些未受邀请的玩家)以某种聪明的方式挫败游戏规则。违背规则的企图是为了获得某种未授权的优势, 例如造成秘密信息的泄露、改变数据而不被发现、伪造证据、责任否认、破坏审计和信任、降低可用性或不提供服务等。现代通信在事务处理、商业运作、提供服务(以及很多其他方面, 如公司业务、个人信息、军事活动和国家事务的保密)方面的重要性意味着要求不遵守游戏规则的玩家不应该获得任何未授权的优势。

在简单“电话掷币”密码协议的开发中, 我们已经目睹了如下过程: 容易破坏的通信游戏演变成一个密码协议, 而且能完成所希望的安全服务。我们的例子展示了密码技术在维护“玩游戏”秩序方面的作用。的确, 在开放的计算机和通信网络中保证安全通信, 采用密码学是一种有效而且是惟一可行的办法。密码协议正是以密码技术“武装”的通信程序, 因而具有保护功能, 保持通信的正常秩序。电子商务、事务处理和服务对安全通信的无尽需求, 再和另一种对“不按照规则游戏”的不停诱惑的企盼的需求, 结合在一起就导致了很多人密码系统和协议的产生, 这些系统和协议构成了本书的主要内容。

## 1.2 描述密码系统和协议的准则

我们应当从一个基本的问题开始:

什么是好的密码系统和协议?

显然,这个问题不好回答! 其中一个原因是由于“好”一词有很多意思,不同的意思也就有不同的回答。本书的一个主要任务就是对这个基本问题给出更深入的答案。但本章我们只能给出一些初步的答案。

### 1.2.1 保护的程度与应用需求相符合

让我们来看我们在 1.1.1 节中所设计的第一个密码协议。

我们可以说“电话掷币”是一个很好的协议,因为从概念看它非常简单。一些已经熟悉很多实用单向杂凑函数(hash function,例如 SHA-1,见 10.3.1 节)的读者,也许会更进一步想到, $f(x)$ 是很容易计算的,即便在袖珍计算器上也不难实现。例如,SHA-1 的输出是 160 位的比特串,或 20 字节;采用 16 进制的编码方案(见例 5.17),这样的输出可编码成 40 个十六进制的字符<sup>①</sup>。因此,Alice(Bob)在电话上读(草记下)上述数据不至于太令人厌烦。这样一种实现方式对 Alice 和 Bob 要决定娱乐场点来说应该说是足够安全的:如果 Alice 想欺骗,她就要找出  $x \neq y \pmod{2}$  使  $f(x) = f(y)$ ,而这将是一个异常困难的问题;同样,Bob 也将面临一个异常困难的问题,即已知  $f(x)$  判断  $x$  是奇数还是偶数。

不过,我们判断采用 SHA-1 来实现“电话掷币”协议的质量是根据游戏人对游戏的结果的认真程度来说的。但在很多更重要的应用场合(例如,1.2.4 节中要讨论的情况),要将公平掷硬币的原型用于密码,所要求的单向性和承诺性一般要比实用的单向杂凑函数如 SHA-1 所能提供的要强得多。应该注意,如果我们仅从字面上理解“不可能”这一概念,具有性质 1.1 的函数就是一个完全安全的单向函数。这种函数是不容易实现的。更糟糕的是,甚至其存在性仍然是一个公开问题(尽管我们对其存在性是乐观的,在 1.1.2 节中可以看到我们的乐观看法,我们将在第 4 章进一步讨论单向函数的存在条件)。因此,对很多公平掷币的更重要的应用场合,实用的杂凑函数不一定就是好的,必须用更加严格的密码技术。另一方面,对决定娱乐场点来说,使用重量级密码技术显然是不必要的。

应当指出,对有些应用来说,太强的保护甚至会妨碍想要的安全服务的正常性能。例如,Rivest 和 Shamir 提出了一个称做 MicroMint 的微支付方案[244],它利用了一个众所周知的、低效率的加密算法来实现方案的优势。该支付系统采用了如下的合理假设:只有资源充足的服务提供商(如大银行或金融机构)才能在一个实用的单向函数下以较低代价找出大量的“碰撞”。也就是说,服务提供商可以计算出  $k$  个不同的数  $(x_1, x_2, \dots, x_k)$ ,使得

$$f(x_1) = f(x_2) = \dots = f(x_k)$$

$x_1, x_2, \dots, x_k$  被称为单向函数  $f$  下的碰撞。很容易验证一对碰撞,因为计算它们的单向函数是很容易的,这些碰撞可以看成是资源充足的服务提供商发布的,因而可表示一个被证实的

<sup>①</sup> 十六进制字符是集合  $\{0, 1, 2, \dots, 9, A, B, \dots, F\}$  中的元素,用来表示 4 比特数的 16 种情况。

值。人们建议用数据加密标准(DES,见 7.6 节)作为实现这种单向函数的适用算法[244],相应得到的输出空间不大(64 位)。因此,并不像平常密码学使用的单向函数那样,其中一个碰撞几乎就构成对系统的一次成功攻击(例如,“电话掷币”协议例),在 MicroMint 中,碰撞是用来实现一个奇妙的微支付服务!显然,利用大得多的输出空间的强单向函数(即 $\gg 64$  位,如有 160 位的 SHA-1),即便对一个资源充足的服务提供商,也不能提供任何服务(在 3.6 节,我们将研究寻找 hash 函数碰撞的计算复杂度)。

虽然采用重量级密码技术来设计安全系统(例如多层加密、任意使用数字签名、由一个甚至多个可信赖第三方提供在线服务)可以使人们觉得能实现更强安全性(也可能减轻设计工作),但这种感觉常常只会提供虚假的保证。人们并不希望穿上累赘的“盔甲”来表演“杀鸡用牛刀”,因为这样做可能会要求更强的安全性假设,并导致更加复杂的系统。复杂的系统还会增加安全性分析和安全实现的困难(因而更容易出错),以及在运行和维护上更高的费用支出。

设计密码或安全系统的一项更重要和更具挑战性的工作是,仅采用必不可少的技术来实现足够的安全保护。这也是衡量一个密码和安全系统好坏的一个重要因素。

### 1.2.2 对安全性的信心要依据所建立的“种系”<sup>①</sup>

我们如何才能相信一个密码算法或协议是安全的呢?能否因为还没有人攻破它就说它是安全的呢?令人遗憾的是,答案是否定的。一般地,对一个未被攻破的算法,我们只能说还不知道如何攻破它。因为在密码学中,一个被攻破的算法有时意味着能够定量地度量;如果一个未被攻破的算法没有这种度量,那我们就甚至无法断言一个未被攻破的算法是否比一个已被攻破的算法更安全。

然而,也有一些例外。在大多数情况下,攻破一个密码算法或方案的任务归结为解决某个数学问题,如方程求解或函数求逆。这些数学问题被认为是“困难的”或“不可解的”。“困难的”或“不可解的”的形式化定义将在第 4 章中给出。这里,我们可以非正式而保险地说,如果一个数学问题尚不能用已知的方法在合理的时间段内求解,则它是困难的。

有很多著名的困难问题常被作为现代密码学的标准组成成分,特别是在公钥或非对称密码体制中(见 8.3 节~8.4 节)。例如,在公钥密码学中,困难问题包含大整数分解问题、离散对数问题、Diffie-Hellman 问题,以及其他一些有关问题(在第 8 章中我们将定义和讨论这些问题)。这些问题可以被看做为已建立的零散“种系”,因为它们已被一代又一代的数学家长期不懈地研究,所以有很高的可信度令人相信,要解它们确实很困难。

今天,要建立对密码算法安全性的高可信度,其标准技术就是给出一个形式化的证明,能够展示对预算法一种攻击就可用来解决“种系”中的一个困难性问题。证明本身就是一个或一系列高效的数学变换,从对算法的一种攻击变为困难问题的一种解法。这种高效变换称为归约,即把对算法的攻击“归约”为解困难问题。由于我们深信归约成的困难问题很可能是无法解的(特别是在以攻击和归约变换来度量的时间代价上),我们将能推出一种可度量的信任度,从而能断言:所谓的攻击是不存在的。这种安全性证明方法因而被称为“矛盾归约”:某个困难问题是易解的。

---

<sup>①</sup> 此处指密码算法源于“名门望族”,如大整数分解、离散对数等公认的数学难题。

形式化的可证明安全性,特别是在各种强力攻击即适应性攻击(Adaptive Attack)模型下,是评价一个密码算法和协议“好坏”的重要标准。我们将采用适用安全性(Fit-for-application)来命名在各种强力攻击模型下通过形式化和矛盾归约方法所建立起来的安全性的质量。

作为本书的一个重要议题,我们将研究很多密码协议和算法的适用安全性。

### 1.2.3 实际效率

我们说一个数学问题是高效的或高效可解的,是指该问题能在问题规模的多项式时间内可解。第4章中将给出的效率的形式化定义,使我们能对这类断言提供精确的度量。

我们暂时不考虑断言的定量上的细节。可以粗略地说,这类断言将所有的问题分成如下两类:容易处理的和困难的。这种划分在奠定现代密码学(基于计算复杂性理论的密码)的基础中起着十分重要的作用。显然,一个密码算法一方面应该设计成容易处理的,以便能被合法用户使用;另一方面还应该是困难的,以便对非用户和攻击者构成一个要解的困难问题。

不过我们应该注意到,上述可解性的断言覆盖了很宽的量度范围,对一个合法用户来说,如果一个问题的计算时间是一个很大的多项式,那么其“效率”一般认为是不可行的,即无任何实用价值。因此,密码算法是“好的”的一个重要准则是,对合法用户来说它应该是实用高效的。特别地,衡量用户资源代价的多项式应该很小(即为低次式,第4章将介绍这一概念)。

在第14章,我们将讨论有关可证明强公钥密码体制的一些先驱性工作。这些工作所提出的一些公钥加密算法,都是在一个共同的动机下提出的,很多公钥加密算法的基本形式是不安全的(我们把这些不安全的方案称为“教科书式密码”,因为大多数密码教科书所介绍的就是这种基本的和初始的形式,本书第III部分将给予介绍)。可是,关于可证明强公钥密码系统的大多数初创性工作都采用逐位加密方式[127,212,243],有些甚至采用超常的步骤和公钥认证框架[212]来增进对每一比特加密正确性的知识证明[243]。虽然这些早期的先驱工作对达到强安全性提供深入看法上是重要的,但他们所建议的系统一般都效率低下而不实用。在第14章之后,我们将进一步研究继先驱工作之后出现的一系列有关可证明强安全的公钥密码体制和数字签名方案的工作。这些后来的研究工作所提出的密码体制不仅具有强安全性,而且在效率方面也是实用的。它们确实是非常好的密码体制。

一个密码协议不仅是一个算法,也是一个通信过程,该过程包含不同协议参与者在一组约定规则下通过计算机网络进行的消息传递。因此,在效率度量方面协议就多了一个量纲:通信交互的次数,通常称为通信轮数。一般地,通信的一步比一步本地计算(典型的一组计算机指令集的执行,例如计算设备上两个数的乘法)的代价要大。因此,人们希望尽量减少密码协议的轮数。评价一个算法是否高效的标准有效性准则是,其运行时间以问题规模的低次多项式为界。如果我们把这个有效性准则用到协议上,那么一个高效协议的轮数应以次数很低的多项式界定:常数(0次)或至多为线性函数(1次)。一个协议的通信轮数超过线性函数,它就不会被认为是实际有效的,即实际中毫无用途。

在18.2.3节中,我们将讨论一些通信轮数以非线性多项式度量的零知识证明协议。应该指出,那些协议并不是为现实应用提出的,但它们在密码和计算复杂性理论方面很重要。对于设计实用高效的零知识协议,我们将在第18章看到更多的研究工作。

### 1.2.4 采用实际的和可用的原型和服务

适用于某种应用的安全性级对于另一种应用未必就足够好。让我们再次以掷硬币协议为



例。在 1.2.1 节中我们已经看到,如果采用一个实用的单向杂凑函数来实施“电话掷币”协议,那么对 Alice 和 Bob 通过电话来决定娱乐场点来说,该协议已经够了。但是,在公平掷硬币原型的许多密码应用中,安全服务都在严格得多的安全级上抗击欺诈和/或实现公平性;在一些应用中,要求绝对意义上的严格性。

例如,在第 18 章我们将讨论一个零知识证明协议,它需要随机比特串输入,且这个随机输入必须得到证明者和验证者双方的信任,否则会对一方或双方造成严重损害。在这种零知识证明协议中,如果通信双方不曾接触或不信任基于第三方服务所提供的随机数(这种服务常被戏称为“天上掉下来的随机数”来意指其不现实),那么他们就不得不通过掷币协议,交互式地逐比特产生大家都信赖的随机数。注意,这里以(通过掷币协议)逐比特来产生随机数的方式满足某种特定要求,如协议的正确性和零知识性。在这种情况下,仅有实用上好这一安全级(即在“电话掷币”协议中采用实用杂凑函数的意义上)很可能不能满足要求。

在密码学和密码协议的应用性研究中,一项具有挑战性的工作就是从实用和可得到的密码原型建立高质量的安全服务。让我们再次引用掷币协议来阐述这一观点。这是由 Blum 提出的一个远程掷币协议[44]。Blum 协议利用实际安全和容易实现的“单向”函数,按非常流行的方式实现了高水平的安全性,可做如下解释:

- 首先,它对抗击掷硬币方(如 Alice)欺诈的困难性实现了定量的度量,即找出一对碰撞  $(x, y)$ , 满足  $f(x) = f(y)$  而  $x \neq y$ 。这里,困难性被量化为分解一个大的合数,即解决一个“种系”困难问题。
- 其次,猜测方绝对无法以超过  $1/2$  的概率获胜,这可用完全安全性来称颂。

因此,从仅使用实用密码的原型就能实现强安全性的意义上来说,Blum 掷币协议是非常好的。作为我们的第一个密码协议的强化和具体实现,我们将描述 Blum 掷币协议,这是本书的最后一个密码协议。

公钥密码学[98,99,248]出现几年之后,大家逐渐认识到,一些基本的且最著名的公钥加密算法(我们称之为“教科书式密码”)一般有两种弱点:(i)它们会泄露加密的消息的部分信息;(ii)它们对主动攻击都相当脆弱(见第 14 章)。这些弱点意味着“教科书式密码”不适于实际应用。早期对教科书式密码弱点通常的补救方法一直是采用逐比特加密方式,甚至应用逐比特的零知识证明技术加上认证框架来防止主动攻击。这些结果虽然对可证明安全的公钥加密算法的发展有价值,但对大多数加密应用来说不适用,这是因为在加密算法中要求零知识证明或认证框架是不切实际的。

自从采用随机化填充方案强化公钥加密算法获得初步成功以后[25],产生了一种利用诸如杂凑函数和伪随机数生成器等通用原型,将普通教科书式公钥加密算法强化为可证明安全算法的一般方法。这些强化的加密方案是实用的,因为它们使用了像杂凑函数等实用的原型,因此其效率与相应的“教科书式密码”类似。由于这一重要的质量因素,一些使用实用和通用的原型强化后的算法成为公钥加密和数字签名的标准。我们将在第15章和第16章研究其中的一些方案。

利用已有的和通用的技术和原型来设计密码体制、协议和安全系统,从如下的意义上说,也是所希望的:由于这些结果能吸引广泛的兴趣来对其进行详细审查,最终它们很可能是安全的。

### 1.2.5 明确性

在 20 世纪 60 年代末期,软件系统变得非常庞大和复杂,计算机程序员开始遇到危机,即所



谓的“软件危机”。大而复杂的软件系统越来越容易出错,排除错误的代价远远超过程序设计和开发的费用。不久,计算机科学家发现了一些制造危机的“罪犯”,即坏的编程习惯,包括:

- 乱用 GOTO 语句(上下跳转看起来非常方便)
- 大量使用全局变量(造成无法控制它们的值的变化,例如在非预期的子程序的执行中)
- 不声明变量类型就使用(隐含类型可以在 Fortran 中使用,例如,一个实数可以截短为一个整数而没引起程序员的注意)
- 非结构化的、组织混乱的大块代码用于多个任务(一块可能有上千行语句)
- 注释行太少(由于它们并不被执行!)

程序员编程时可能图“方便”,但已经证明,这种做法将给程序的调试、维护和进一步开发带来很大的困难。用这种“方便”特征所设计的软件代码可能太含糊不清而难以理解和维护。一个程序员在几个月或几个星期前写的一小段代码,现在却看不懂了,这种现象经常出现。

一旦理解了坏的编程习惯会带来灾难性的后果,程序设计方法学也就成为一个研究学科,在这个学科中,明确性是编程的一个重要原则。明确性包含限制使用 GOTO 语句和全局变量(最好是完全不用);明确声明(通过强制)任何变量的类型,使得编译器能系统地和自动地检查类型错误;模块化程序设计(把大的程序分解成许多更小的部分,每部分执行一个任务);使用足够的(尽量清楚)说明性材料,这些材料是在程序中或文档外以文本文件的形式出现的。

一个安全系统(密码算法或协议)包括软件和/或硬件中实现的程序部分,对于协议的情况,程序的各部分要在很多独立的主机上运行(或很多程序并行、交互地运行在这些主机上)。软件工程的明确性原则自然适用于安全系统的设计(特别是对协议)。可是,安全系统是假定运行在一个充满敌意的环境下的,即使合法用户在该环境下也可能是恶意的。因此这种系统的设计者还应该明确很多的附加问题。这里我们列出三个重要方面来作为对安全系统的设计者和实施者的一般指导原则(在本书的其余部分,我们将遇到很多对算法和协议的攻击,它们源于这些系统的设计或说明不够明确)。

### 1. 要明确所需要的所有假定

一个安全系统要通过和环境的交互来运行,因而有一组由所在环境必须满足的要求。这些要求称之为系统运行的假定(或前提)。违反一个协议的假定就可能招致对系统的攻击,其结果可能是使假定的服务成为泡影。违反一个没有特别清晰说明的(隐含的)假设是很难被发现的,因此安全系统的所有假定都应当非常明确。

协议中有一个隐含的假定或期望是很平常的,例如假定或期望协议所运行的主机可以提供好的随机数,但现实中的台式机或掌上设备很少能满足这一假定。如果采用很差的随机数源,就可对其协议实施一种所谓的低熵攻击。对安全套接层(SSL)协议(WWW 浏览器与服务器间的一个认证协议,见 12.5 节)早期实现的一种广泛发布的攻击就是低熵攻击的一个著名例子[125]。

对假定的明确鉴别和规范也有助于对复杂系统的分析。DeMillo 等([92]的第 4 章)、DeMillo 和 Merritt[93]建议密码协议的设计和分析采用下面的两步方法(Moore[206,207]对其改进后):

- i) 标明协议中所有的假定。
- ii) 对第(i)步的每一个假定,确定违反该假定对协议的安全性所造成的影响。

## 2. 要明确所提供的确切的安全服务

密码算法/协议提供特定的安全服务,一些重要的安全服务包括:机密性(消息不能被非接收方理解)、认证性(能确定消息的完整性或消息来源)、不可否认性(不能否认对消息的接触)、知识证明(在不泄露的情况下出示证据)和承诺(例如,我们的第一个密码协议——“电子掷币协议”所提供的服务,Alice 无法改变曾使用的字串)。

当设计一个密码协议时,设计者应该对协议要提供什么样的服务非常明确,而且对这些服务应该进行明确的说明。这种明确鉴别和规范不仅能帮助设计者选择正确的密码原型或算法,而且也能帮助实施者正确地实现协议。服务的鉴别对于上述例子中所给定的一般服务的级别细化往往是不合用的,还需要对它们进一步细化。下述是一些可能的改进方法:

机密性(Confidentiality)	⇒ 隐私性、匿名性、不可见性、不可区分性
认证(Authentication)	⇒ 数据源、数据完整性、对等实体
不可否认性(Non-repudiation)	⇒ 消息发布、消息收据
知识证明(Proof of Knowledge)	⇒ 知识拥有权、知识结构

在协议设计中,对服务的错误鉴别可能导致误用密码原型,从而导致协议的安全性缺陷。在第2章和第11章我们将看到灾难性的例子,由于对安全服务在机密性和认证性的错误鉴别,导致了认证协议的严重安全缺陷。

还可能有很多名字更特别的安全服务(例如,消息的新鲜性、不可延展性、前向保密、完善零知识、公平性、绑定性、否认性、无收据等)。这些服务可被看成是从前面已列出的一般性服务的导出结果或进一步细化(导出也可以是否定的,例如否认性就是不可否认性的否定导出)。尽管如此,为了避免设计缺陷,对它们进行明确区分往往是必要的。

## 3. 要明确数学方面的一些特殊情况

正像在1.2.2节中讨论过的那样,计算复杂性理论中的一些困难问题能给密码算法或协议提供更高可信度的安全性。但是,一个困难问题往往有一些特殊情况,其中的该问题根本不难。例如,大整数分解问题一般是很困难的,但对大合数  $N = PQ$ ,如果  $Q$  是大素数  $P$  的下一个大素数,则分解就不是什么难题! 我们可以通过计算  $\lfloor \sqrt{N} \rfloor$  ( $\lfloor \cdot \rfloor$  称为底函数,表示取整)并用一些离  $P$  和  $Q$  很近的素数试着去除就能很快求解。

密码算法赖以工作的常用代数结构(如在第5章将研究的群、环和域)也包含一些特殊情况,也会产生一些例外的容易问题。乘法群或有限域中的低阶元素(定义见第5章)就提供这样一个例子,一个极端情况是 Diffie-Hellman 密钥交换协议(见8.3节)中的基是这些代数结构中的单位元。椭圆曲线的弱的情况,例如,“超奇异曲线”和“非正规曲线”也是很好的例子。超奇异曲线上离散对数问题可归约为有限域中的离散对数问题,称为 Menezes-Okamoto-Vanstone 攻击[199](见13.3.4.1节)。“非正规曲线”的点数与相应的有限域中元素的个数是一样的,其离散对数问题存在一个多项式时间攻击,称为 Satoh-Araki[254]、Semaev[260]和 Smart[280]攻击。

一种容易的特殊情况如果未能被算法/协议的设计者搞明白,或/和在算法/协议规范中没有明确说明,那么它就可能很容易地混入到实现中去,从而被攻击者所利用。因而算法/协议设计者必须警觉数学方面的特殊情况,并应该对实现者明确地说明其过程,以避免这种情况的发生。

为明确性而列出更多的条目并不困难(例如,密钥管理协议应该明确地规定密钥管理的规则,比如不同用途的密钥应该分离,适当的密钥分发步骤等)。由于这些条目的特定性,我们无法

在这里把它们都列出来。不过,作为密码算法/协议的设计和规范的一个一般准则,明确性在本书的其余部分将会经常提及。一般,一个密码算法/协议设计和规范得越清楚,对其分析也就越容易,因而也就越有可能正确地实现,算法/协议遭受意想不到攻击的可能性也就越小。

### 1.2.6 开放性

密码学一度为政府专有。军事和外交机构用它来保密消息。在那些年代,大多数密码研究都是关起门来做的,算法和协议是秘密。的确,政府过去而且今后仍然会认为,对他们的密码研究活动进行保密是有道理的。让我们设想一个政府机构发布了一个密码。我们仅能认为所发布的密码是可证明安全的;否则发布就很危险并最终可能让政府感到非常尴尬。那么别的政府可能会利用这个可证明安全的密码,结果就削弱了发布这一密码的政府中破译者的战斗力。

可是今天,密码机制已经与很多大范围的民用系统结合在一起(在本章最开头我们已粗略地列举了一些应用)。民用的密码研究应该采用公开的途径。密码算法确实用到一些秘密,但这些秘密应该局限在密码的密钥或密钥数据(如口令或个人身份号 PIN);算法本身应该公开。让我们来解释这样规定的理由。

在任何研究领域,高质量的研究都依赖于通过会议报告和学术期刊上发表文章来公开交换想法。这种交换以一般的形式进行。但对于密码算法、协议和安全系统,公开研究不仅是一般意义上的获得和增长知识。公开研究的一个重要作用是公开的专家检验。大家都知道,密码协议、算法和安全系统很容易出错。一个密码研究结果一旦被发布,就可以由很多专家来检验。因而设计者所忽视的错误(可能在设计中或在安全分析中)被发现的机会就大大增加了。相反,如果对算法的设计和开发保密,那么为了保密,即使可能有,也只有很少专家能够接触和检查算法的细节,结果发现错误的机会也就减少了。一个更坏的情况是,设计者可能已经知道有错误,并将其秘密地用于谋取私利。

民用的密码算法、协议和安全系统必须公开,并且必须经过长时间的公开检验,现在已经成为一个既定的准则。一个安全的系统也应该由反对的专家进行同等的评价。

## 1.3 本章小结

在本章,我们以应用密码学的一个简单例子开场。这个例子达到了三个目的:

- i) 展示了密码学在解决问题方面的效力
- ii) 致力于对密码学基础的了解
- iii) 强调了非教科书式安全方面的重要性

这些构成了本书其余部分要讨论的主要内容。

我们接着进行了一系列的讨论,目的是给出这一研究领域的初始背景和文化导论。我们对这些方向的讨论并不完全。其他几位作者已对密码学和信息安全领域的原理、指导原则和文化进行了广泛的研究。下面的几本书是很好的进一步阅读材料: Schneier[256]、Gollmann[131]和 Anderson[14]。Schneier 创办的月刊“Crypto-Gram Newsletters”也是很好的阅读材料。要订阅这份快报,请发电子邮件到 [schneier@counterpane.com](mailto:schneier@counterpane.com)。

## 习题

- 1.1 算法和协议的区别是什么?
- 1.2 在协议 1.1 中, Alice 能决定正面或背面。这在某些应用中可能是不公平的, 请修正该协议, 以便使 Alice 不再具有这个优势。  
提示: 让正确的猜测决定是正面或背面。
- 1.3 设函数  $f$  是从 200 比特的整数集合到 100 比特的整数集合的映射, 映射规则如下:

$$f(x) \stackrel{\text{def}}{=} (x \text{ 的前 100 比特}) \oplus (x \text{ 的后 100 比特})$$

这里  $\oplus$  表示逐比特异或 (XOR) 运算, 即

$$a \oplus b = \begin{cases} 0 & \text{若 } a = b \\ 1 & \text{其他} \end{cases}$$

- i)  $f$  的效率高吗?
  - ii)  $f$  是否具有“性质 I”?
  - iii)  $f$  是否具有“性质 II”?
  - iv) 该函数能在协议 1.1 中使用吗?
- 1.4 未被攻破的密码算法比已被攻破的就安全吗? 如果不, 为什么?
  - 1.5 复杂的系统容易出错, 给出复杂的安全系统更容易出错的其他原因。

## 第2章 防守与攻击

### 2.1 引言

存在众多密码学协议的一个原因是基于以下事实:确保密码协议正确是很困难的。设计一个正确的协议要投入无限的精力。许多新提出的协议是为了修补已发现的现有协议的安全缺陷。密码协议中的安全隐患总是可以用一个攻击脚本描述,在其中协议所提供的一些安全服务能够被一个或一些相互勾结的攻击者蓄意破坏。在密码协议领域,似乎永远存在着协议设计者和攻击者之间的较量:提出一个协议,发现一种攻击,跟着进行修补,而后又发现另一种攻击,再一次修补……

在这一章中,我们将演示一系列攻击和修补之间争斗的例子。我们从故意设计一个存在缺陷的做作协议出发,通过将这个协议经历“修补、攻击、再修补、再攻击”的过程,最终将得到两个协议,它们是由计算机安全学家们所设计的,用以解决现实世界中的信息安全问题(此前所有那些有缺陷的和被修补的,因而被攻破的协议都是故意设计的)。这两个源于我们的“修补、攻击、再修补、再攻击……”过程所得到的协议不仅是真实的,而且是闻名的。其理由有二,它们在应用上和和密码学协议的形式化分析的基础性重要研究上都起着根本性的作用。

遗憾的是,这两个通过修补得到的真实协议仍含有安全缺陷,在协议发表很久以后才发现它们。其中一个协议的一个缺陷是发表三年之后才发现的,而另外一个协议的一个缺陷竟然又过了十四年才被揭露!在披露这些缺陷之后,我们将试图对其进行最后的修补。虽然在最后修补结果中还有某些进一步的安全问题,但我们将推迟到后面的章节中再去揭示,那时我们已具备了处理这些问题的能力。本章不打算解决安全问题,而是想提出一个“早期警报”消息:密码算法、协议和系统很容易包含有缺陷。

本章也作为一个材料和理念上的技术性引论,使我们(尤其对于刚接触到密码学、密码协议和信息安全领域的读者)建立起在该领域的研究中遇到的一些常见的重要概念、定义和约定,包括一些基本的术语、它们的含义(第一次出现的词会用黑体形式)和一些在全书中经常遇到的协议参与者的约定取名。还会介绍对这些有安全缺陷的协议的各种攻击,这将使我们对在我们的对弈游戏中扮演特殊角色(攻击我们所设计的密码协议的敌人)的一些典型行为逐渐熟悉起来。

#### 2.1.1 本章概述

在2.2节我们介绍了只用于本章的一个简化的加密概念。在2.3节~2.5节中我们将介绍密码技术,特别是认证、协议中的标准威胁模型、环境和目标。最后,在2.6节我们对一系列认证协议展开讨论。

### 2.2 加密

本章中要设计的所有协议都用到**加密**技术。我们要对这种“一锅煮”的加密做法及早提出

一个忠告:在许多情况下,这种用法是不正确的,而应该用其他一些密码原型来代替。在本书中,我们将逐步建立对为获得精确的安全服务需要准确地应用密码原型的意识。不过,为了简化我们的介绍,在本章中我们仅使用加密技术。

加密(有时称为**加密作业**,是将一条信息变换成为不可理解的形式过程。输入到该变换器的信息叫**明文**或**明报**,输出的叫**密文**或**密报**。将密文转换成明文的过程叫**解密**或**解密作业**。注意明文和密文是一对相对应的记号:前者是指一个加密算法的输入,后者是指输出。明文不必是可理解的;例如,在双重加密的情形中,位于中间的密文可以看成是第二次加密的明文;本章我们还会多次看到,在密码协议中对随机数的加密是很普遍的。通常,明报是所有消息集合的一个小的子集,这个小子集具有一定的可辨认的分布。在3.7节中我们研究消息的分布。

加密和解密算法统称为**密码算法**(**密码系统**或**密码体制**, Cryptosystem),加密和解密过程均由密钥控制。在**对称(或共享密钥)**密码体制中,加密和解密采用同一密钥;而在**非对称(或公钥)**密码体制中,加密和解密采用两个不同的密钥:加密密钥和(相匹配的)解密密钥,加密密钥可以公开(因此也称为**公开密钥**)而不会泄露解密密钥(因此公钥密码体制中解密密钥也称为**私钥**)。图2.1给出了密码体制的简单图形描述。更加完整的密码体制的描述将在第7章给出(见图7.1)。

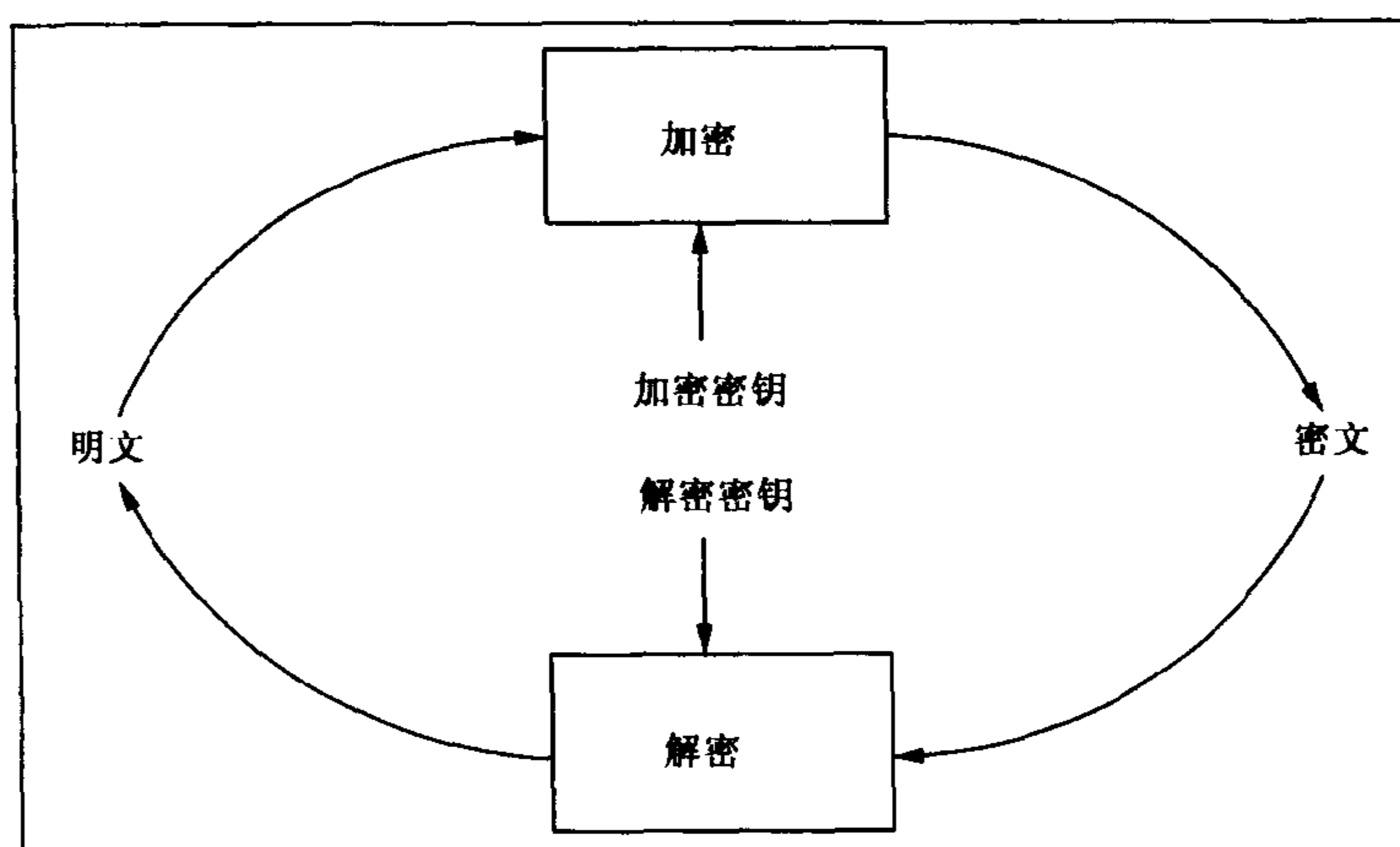


图 2.1 密码系统的简化图形描述

应当指出,在本章范围内,词语“明文”、“密文”、“加密”、“解密”、“加密密钥”和“解密密钥”是成对关联的概念。对于消息  $M$  (不管是明文还是密文)、密码算法  $A$  (不管它代表加密还是解密)和密钥  $K$  (不管它是加密密钥还是解密密钥),我们可以用

$$M' = A(K, M)$$

表示一个**密码变换**,它代表了图2.1中上面方框或者下面方框的作用。因而,我们可以用  $A'$  和  $K'$  来表示

$$M = A'(K', M')$$

即

$$M = A'(K', A(K, M))$$



以走完图 2.1 中的圈。在对称钥密码体制中,我们可以认为  $K' = K$ ,在公钥密码体制中, $K'$  表示  $K$  的秘密部分或与  $K$  相匹配的部分。本章协议中消息的密文习惯上记为

$$\{M\}_K$$

在学了消息的概率分布(在 3.7 节~3.8 节中介绍)后,我们就会知道明文(更精确地说,明报或可懂的)消息是整个消息空间中的一个小的子集,而密文消息在这个空间中的分布要广得多,这是明文和密文的本质区别。

我们应该注意,在这一章中,我们所谓的密文总是指采用下述两个意义上的“完善”密码算法所得到的:

**性质 2.1** 用符号  $\{M\}_K$  表示的完善加密

- i) 不用密钥  $K$ (在对称密码体制中),或者不用与  $K$  相匹配的私钥(在公钥密码体制中),密文  $\{M\}_K$  不提供任何求解明文消息  $M$  的密码分析方法。
- ii) 密文  $\{M\}_K$ ,也许还有一些关于明文消息  $M$  的已知信息,不提供任何求解密钥  $K$ (在对称密码体制中)或与  $K$  相匹配的私钥(在公钥密码体制中)的密码分析方法。

具有这两个性质的完善加密(还有一个附加性质将在 2.6.3 节中介绍)是对现实世界中存在的加密算法的理想化。这个理想化能便于我们处理,可将协议设计与分析的责任同协议所使用的密码算法的设计和分析的责任分离。这种分离使协议的设计和分析工作变得容易。我们马上就会看到,完善加密并不能防止协议含有安全缺陷。事实上,本章要示范的协议的任何一种攻击,都不是因为相应的密码体制不完善。

我们将在后面的章节中(第 7 章、第 8 章、第 13 章和第 15 章)介绍加密的形式化定义和几个加密算法。尽管如此,这里对加密和解密作用描述的抽象程度能够满足本章的要求。把加密算法看成是一个有钥匙的挂锁,而把密文看成是装在挂锁锁上的盒子中的文本,对我们目前来说是无害的。

读者可以参阅文献[268]来获得有关信息安全的有用词汇。

## 2.3 易受攻击的环境(Dolev-Yao 威胁模型)

由计算机、设备和资源构成的大型网络(例如因特网)是典型开放性的,这就意味着一个主体(或实体、代理、用户)能够加入这样的网络,并通过该网络来发送和接收消息,而不需要由“超级”主体授权,这里的主体可以是一台计算机,一个设备,一些资源、服务的提供者,一个人或一个组织等。在这样的开放性环境中,我们必须想到会有坏人(攻击者、对手、敌人、入侵者、窃听者、冒名顶替者等),他们会做各种坏事,不仅仅是被动地窃听,而且会主动地改变(可能用某些未知的运算或方法)、伪造、复制、改变路由、删除或注入消息。注入的消息可能是恶意的并对接收端主体以破坏性的影响。在密码学文献中,这些坏人叫主动攻击者。本书中,我们把攻击者称为 **Malice**(是指经常带着不同身份面具做坏事或捣蛋的人)。**Malice** 可以是单个的,也可以是相互勾结的攻击团伙,一个特例是,它可能是协议中的一个合法主体(内部人员)。

一般假定 **Malice** 在操纵经开放性网络的通信方面是相当聪明的。由于他们的行为是不规范的,其操纵技术也是难以预料的。而且,由于 **Malice** 可以表示一个相互勾结的坏蛋团伙,他



可以同时控制地域上相距很远的一些网络节点。Malice 何以能做这些的真正原因将在 12.2 节中介绍。

在这样脆弱的环境中预料到如此强大的对手, Dolev 和 Yao 提出了一个**威胁模型**, 这一模型被广泛地采纳为密码协议的标准威胁模型 [102]。在这个模型中, Malice 有如下特征:

- 他能获得经过网络的任何消息。
- 他是网络的一个合法使用者, 因而能够发起与任何其他用户的对话。
- 他有机会成为任何主体发出信息的接收者。
- 他能够冒充任何别的主体给任意主体发消息。

因此, 在 **Dolev-Yao 威胁模型** 中, 发送到网中的任何消息都可看成是发送给 Malice 处理的 (根据他的计算能力)。因而从网络接收到的任何消息都可以看成是经过 Malice 处理过的。换句话说, 可认为 Malice 已完全控制了整个网络。事实上, 将开放性网络看成是 Malice 是无害的。

但是, 除非明确声明, 我们并不认为 Malice 是全能的。这意味着即使把他看成是一个相互勾结的犯罪团伙, 能够并行使用跨越开放网络中的大量计算机, 也还有一些 Malice 所不能做的事情。下面在不对“不能做”的意思量化的情况下, 我们列出一些 Malice 不能做的事情, 准确的量化将在第 4 章给出:

- Malice 不能猜到从足够大的空间中选出的随机数。
- 没有正确的密钥 (或私钥), Malice 不能由给定的密文恢复出明文; 对于完善加密算法, Malice 也不能从给定的明文构造出正确的密文。
- Malice 不能求出私有部分, 比如, 与给定的公钥相匹配的私钥。
- Malice 虽然能控制我们的计算和通信环境的大量公共部分, 但一般他不能控制计算环境中的许多私有区域, 如访问离线主体的存储器。

我们将采用 Dolev-Yao 威胁模型来处理所有我们将遇到的协议。

## 2.4 认证服务器

假设两个主体 Alice 和 Bob (我们在第一个密码协议“电话掷币”中已遇到过, 协议 1.1) 希望相互通过安全方式进行通信。假设 Alice 和 Bob 以前从未见过面, 因而他们不能事前就有一个共享密钥, 也不能确定对方的公钥。那么他们怎样通过根本不安全的网络安全地通信呢?

直接的方式是 Alice 和 Bob 可约会至少彼此见一面, 并建立一个共享秘密钥或交换有关对方公钥的确切知识。然而, 在有  $N$  个用户希望进行秘密对话的系统中, 为了安全地建立这些密钥, 这些用户需要进行多少次旅行才行呢? 答案是  $N(N-1)/2$ 。遗憾的是, 对于一个庞大系统, 这意味着其代价是不可行的。因此, 在现代通信系统中这种直接的安全密钥建立方式是不实际的。

然而, 每一个主体选用安全的通信来获得**认证 (和目录)**服务是可行的。Needham 和 Schroeder 提出这种服务可以由**认证服务器**来提供 [215]。这种服务器就像一个名字注册机构, 它根据它所服务的主体的名字维护一个检索数据库, 并能够递送通过请求主体密钥所计算出来的身份信息, 而该密钥为服务器和主体所共享。

认证服务器是一种特殊的主体, 它的行为老实并必须得到用户 (客户主体) 们的信赖。也就是说, 只要客户主体要求, 它就严格按照协议规范做出反应, 而不会参与其他任何故意破坏

客户安全的活动(比如,它永远不会向任何第三者泄露它和顾客共享的密钥)。这样的主体称为**可信第三方**或简记为 **TTP**,在本书中我们用 **Trent** 来命名可信第三方。

我们假设 Alice 和 Bob 使用由各自的认证服务器提供给他们们的认证服务。在扩大的网络中,仅用一个中心认证服务器是不明智的。Needham 和 Schroeder 建议用相互了解的多个认证服务器。因而,仅由一个认证服务器提供服务的主体的名字格式为“认证机构.简单名字”。Diffie 和 Hellman 也曾提出了使用多个认证服务器的想法[98]。

然而,为了简单、清楚地描述本章中的协议,我们假设 Alice 和 Bob 使用同一个认证服务器 Trent。在第 12 章,我们将介绍和分析 Windows 2000 操作系统中的网络认证基础——Kerberos 认证协议[91],它考虑采用多个认证服务器管理多个网络的一般性网络构造。

在同一个认证服务器提供服务下,我们假定 Alice(Bob)和 Trent 共享一个密钥,设密钥由  $K_{AT}$  ( $K_{BT}$ )表示。以后我们将会看到这种密钥称为**密钥-加密密钥**,因为它的用途主要是加密其他密钥。由于建立这样一个密钥的高额花费,它应该被长期应用,因此它也叫**长期密钥**。

## 2.5 认证密钥建立的安全特性

本章中描述的所有协议都属同一类型,它们都用来实现认证的密钥建立。这种安全服务的确切含义可由以下的三个性质详细说明。

假设  $K$  表示 Alice 和 Bob 要建立的共享密钥,在本章中所要设计的协议都应实现具有以下三个性质的安全服务。协议执行完毕后:

1. 只有 Alice 和 Bob(或者可能还有他们都信任的某个主体)能够知道  $K$ 。
2. Alice 和 Bob 应当知道对方主体知道  $K$ 。
3. Alice 和 Bob 应当知道  $K$  是新生成的。

第一个条性质是由认证的最基本含义得出的:识别将要通信的目标主体。如果启用密钥  $K$ “加锁”,Alice(或 Bob)应当确信通信的另一端一定是 Bob(或 Alice)。如果密钥建立服务是在 Trent 的帮助下实现的,那么应当相信 Trent 不会伪装成两个主体中的任何一个。

第二条性质将认证服务扩展到另一个方面,即**实体认证**,或者是已识别出的通信对象主体的**活现性**。Alice(或 Bob)应当确信,当前通信协议运行中 Bob(或 Alice)参与了,并对通信做出响应。以后我们将会看到,为了防止对旧消息的重放攻击,这条性质是必要的。

对第三条性质的需求源于密码学中长期确立的**密钥管理**原则。原则规定:当一个密钥是共享密钥并用于大量的数据加密时,只能使用较短的时间。这种密钥的用法和“密码加密密钥”或 2.4 节末所提到的长期密钥的用法大为不同。这种密钥管理原则有两个原因:第一,如果一个数据的加密密钥是共享的,那么即使共享者中的一方 Alice 在密钥的管理和使用中非常谨慎,而在 Alice 的控制之外,由于另一个共享者 Bob 的不谨慎而泄露了这个共享密钥,仍然会导致 Alice 的安全得不到保障;第二,在保密通信中,通常大多数数据包含(可能是大量)已知的或可以预料的信息或结构。例如,一段计算机程序包含大量的已知词汇,如“begin”、“end”、“class”、“int”、“if”、“then”、“else”、“++”等。这种数据包含大量的冗余(定义见 3.8 节)。这类数据的加密使得密钥成为以寻找密钥和明文为目标的**密码分析**的对象,延长这类数据加密密钥的使用时间将会降低密码分析的困难性。我们还应该考虑到,Malice 有无限的时间来寻找一个旧的加密密钥,并在找

到后把它当做新的密钥来用。已经牢固建立并广为接受的密钥管理原理规定,一个共享密钥仅能用于一次通信会话。因此这种密钥也称**会话密钥**和**短期密钥**。认证密钥建立服务的第三条性质向 Alice 和 Bob 保证,建立的会话密钥  $K$  是新产生的。

## 2.6 利用加密的认证密钥建立协议

现在我们已经做好了设计认证密钥建立协议的准备,要设计的第一个协议只是为了直接实现下列简单想法: Alice 和 Bob, 虽然彼此不认识, 但都认识 Trent 并且分别和 Trent 共享长期密钥, 因而 Trent 能安全地在他们之间传递消息。

### 2.6.1 消息保密协议

由于协议的运行环境是脆弱的, 我们的协议将使用加密技术来防范可能的威胁。在一步一步讨论的最初阶段, 我们将把注意力放在对消息保密的威胁上。

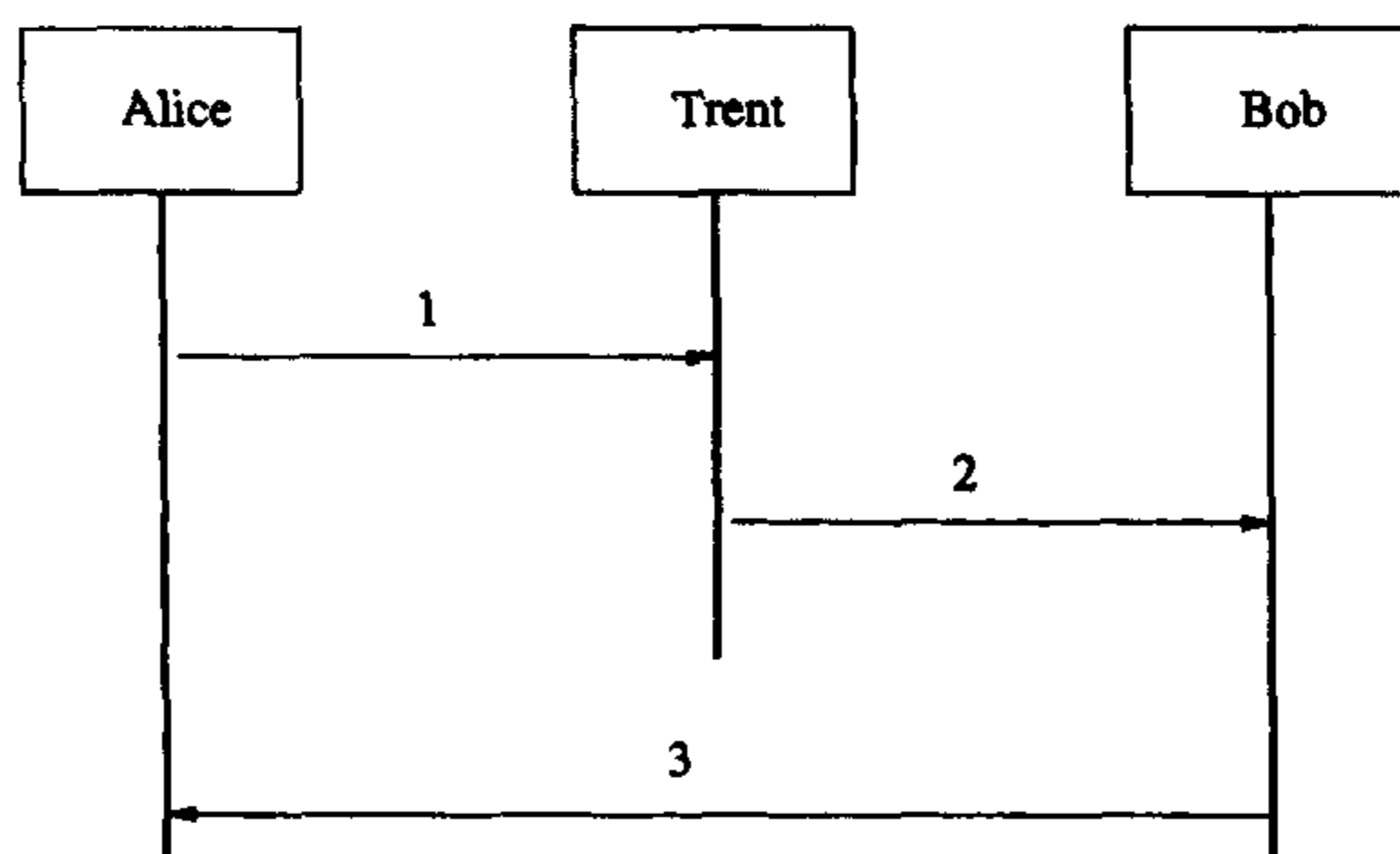
#### 2.6.1.1 “从 Alice 到 Bob”协议

设 Alice 发起运行这样一个协议。她首先随机生成一个会话密钥, 然后用已有的和 Trent 共享的长期密钥对这个会话密钥加密, 并把密文连同自己和 Bob 的身份信息发给 Trent。在收到 Alice 会话密钥递送的请求后, Trent 首先从数据库找出 Alice 请求中提到的两个主体的长期共享密钥, 然后用 Alice 的密钥解密密文, 再用 Bob 的密钥对解密后的结果进行加密并把加密后的新密文送给 Bob。最后, 收到递送的会话密钥数据并经解密后, Bob 就用新收到的会话密钥加密一条消息发给 Alice 来确认此次接收。协议 2.1 示例了实现从 Alice 到 Bob 会话密钥传递的协议描述。在这个协议中, Alice 是**发起者**, Bob 是**响应者**。

#### 协议 2.1 从 Alice 到 Bob

假定 Alice 和 Trent 共享密钥  $K_{AT}$ , Bob 和 Trent 共享密钥  $K_{BT}$ 。

目标 Alice 和 Bob 想要建立一个新的共享密钥  $K$ 。



1. Alice 产生一个随机数  $K$ , 创建  $\{K\}_{K_{AT}}$  并且向 Trent 发送:  $Alice, Bob, \{K\}_{K_{AT}}$ ;
2. Trent 找出密钥  $K_{AT}$  和  $K_{BT}$ , 解密  $\{K\}_{K_{AT}}$  恢复  $K$ , 创建  $\{K\}_{K_{BT}}$  并向 Bob 发送:  $Alice, Bob, \{K\}_{K_{BT}}$ ;
3. Bob 解密  $\{K\}_{K_{BT}}$  恢复  $K$ , 向 Alice 发送:  $\{你好 Alice, 我是 Bob!\}_K$ 。

本章中,我们将分两部分介绍我们的大多数协议(和对协议的攻击),图示部分说明在主体之间传递的消息流,说明部分将给出主体进行发送和接收消息活动的细节。虽然单单说明部分就已足够精确描述协议(在本书其余章节中,仅用说明部分作为协议的描述方式),但我们打算通过增加消息流的图示表示,使密码协议领域的初学者轻松起步。这也是本章的目的之一。

在研究协议“从 Alice 到 Bob”是否包含任何安全缺陷之前,我们应该首先讨论协议的设计特征。协议使得 Alice 生成一个与 Bob 共享的会话密钥,那么 Bob 对此高兴吗? 如果发现由 Alice 产生的会话密钥的随机性并不足够好(密钥应该是随机的以使用猜测很难确定),因为密钥是共享的,那么 Bob 的安全性将不能得到保障。也许 Alice 并不关心会话密钥是否强,或者她只是为了方便记忆。只要 Bob 不信任 Alice(甚至在协议运行前不相识),那么他可能不愿意接受这样一个会话密钥并与之共享。我们将修改这一协议,去掉协议的这一设计特征并讨论修改后协议的安全性。

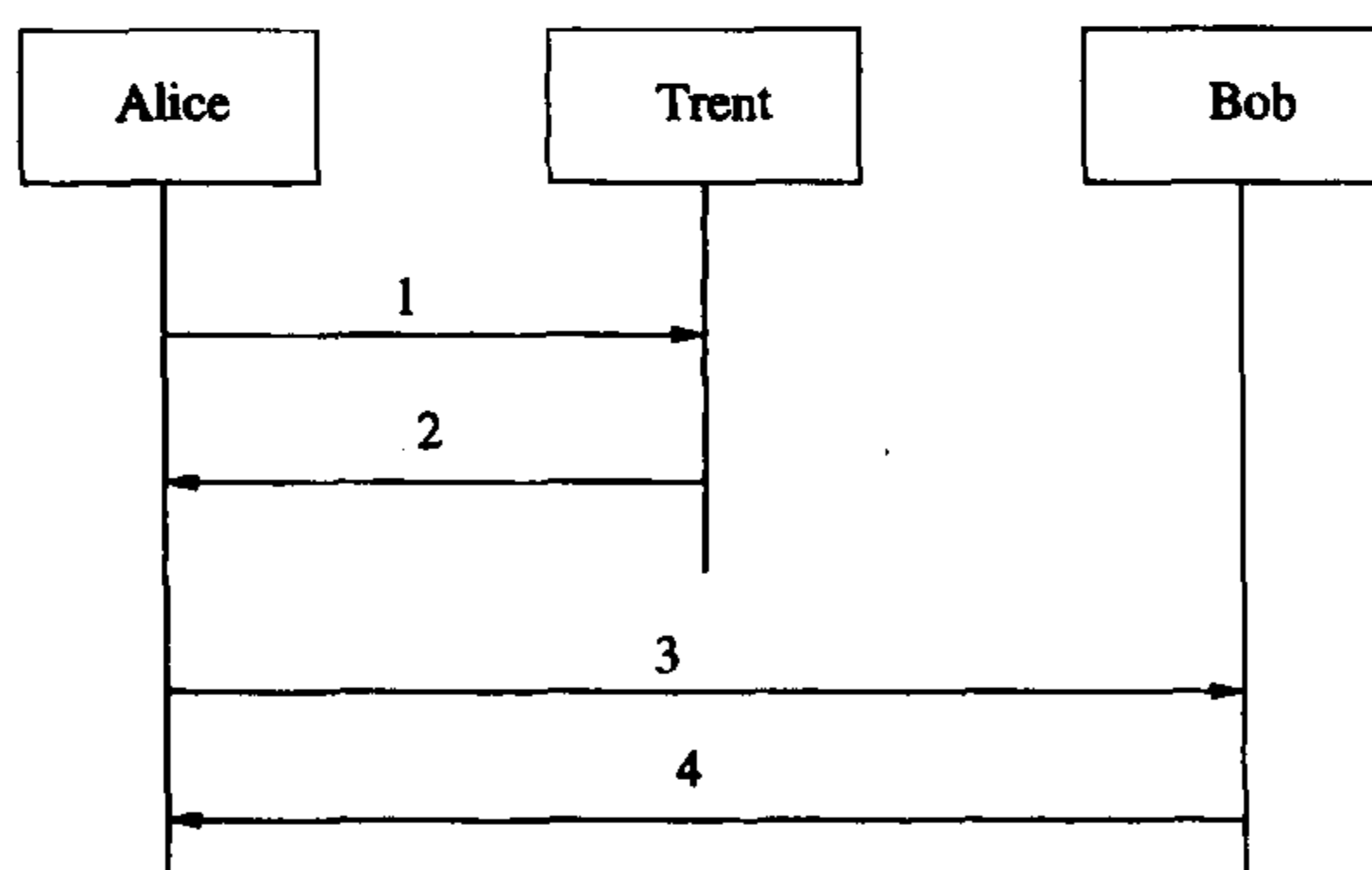
### 2.6.1.2 “来自 Trent 的会话密钥”协议

由于 Trent 得到了两个客户主体的信任,他通过适当方式生成的会话密钥应能得到双方的信任。因而可以把协议 2.1 修改成协议 2.2。首先, Alice 把自己和 Bob 的身份信息发给 Trent, 这两个主体 Alice 和 Bob 想共享一个秘密通信的会话密钥来进行保密通信。在收到 Alice 的请求后, Trent 将在自己的数据库中找到两个主体各自的密钥, 生成一个要在这两个主体之间共享的新的会话密钥, 然后分别用相应的密钥对会话密钥进行加密, 并把加密后的结果送给 Alice。 Alice 对属于自己的部分进行处理, 并把属于 Bob 的那部分传给 Bob。最后, Bob 处理协议中自己的那一部分, 并发送收到会话密钥的一个肯定回执来结束协议。我们把修改后的协议命名为“来自 Trent 的会话密钥”。

#### 协议 2.2 来自 Trent 的会话密钥

假定 Alice 和 Trent 共享密钥  $K_{AT}$ , Bob 和 Trent 共享密钥  $K_{BT}$ 。

目标 Alice 和 Bob 想要建立一个新的共享密钥  $K$ 。



1. Alice 发给 Trent: *Alice, Bob*;
2. Trent 找出密钥  $K_{AT}$ 、 $K_{BT}$ , 随机生成  $K$ , 发送给 Alice:  $\{K\}_{K_{AT}}, \{K\}_{K_{BT}}$ ;
3. Alice 解密  $\{K\}_{K_{AT}}$  并发给 Bob: *Trent, Alice,  $\{K\}_{K_{BT}}$* ;
4. Bob 解密  $\{K\}_{K_{BT}}$  恢复  $K$ , 发给 Alice:  $\{\text{你好 Alice, 我是 Bob!}\}_K$ 。

在完善加密方案下对会话密钥加密后,一个被动的窃听者在看到“来自 Trent 的会话密钥”协议的运行后,因为这些密文仅能由合法的接收者用各自的密钥解密后阅读,在没有加密密钥  $K_{AT}$  和  $K_{BT}$  的情况下,无法得到会话钥  $K$  的任何信息。

### 2.6.2 攻击、修复、攻击、修复……

现在我们来举例说明本书的一个标准场景,即攻击、修复、攻击、修复……

#### 2.6.2.1 一个攻击

协议“来自 Trent 的会话密钥”是有缺陷的。此协议的问题是,谁应该得到会话密钥这一信息没有得到保护。攻击 2.1 给出了一个攻击方式。在这个攻击中, Malice 截获网上传递的一些消息,将它们修改后并假冒某些主体发给另外一些主体。在攻击 2.2 所描述的攻击中,我们用

Alice 发送给 Malice(“Trent”):……

来表示 Malice 截获 Alice 发给 Trent 的消息的行动,用

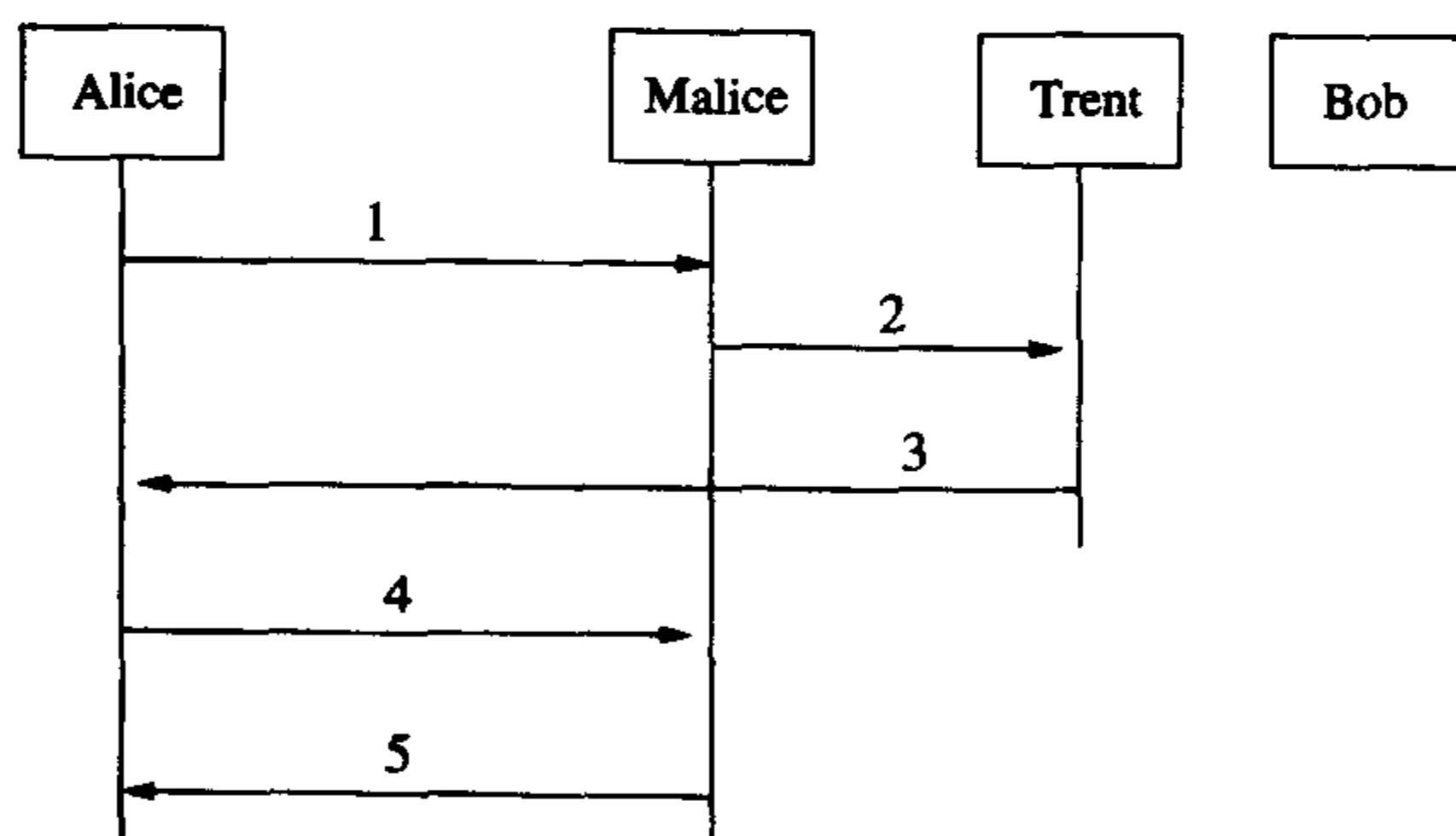
Malice(“Alice”)发送给 Trent:……

来表示 Malice 通过模仿 Alice 向 Trent 发送消息的行为。我们应该注意,根据我们在 2.3 节所同意的协议环境的 Dolev-Yao 威胁模型,假定 Malice 能完全控制脆弱的网络。因此 Malice 能够采取上面提到的恶意行为。我们能够想像,当 Malice 操纵了通过网络的消息时,代号(“主体\_名字”)就成了 Malice 佩戴的面具了。在 12.2 节我们将从技术上看到 Malice 是如何通过这种办法操纵在网络中所传递的消息的。

#### 攻击 2.1 对“来自 Trent 的会话密钥”协议的一种攻击

假定 除了原协议的假定外, Malice 和 Trent 共享密钥  $K_{MT}$ 。

攻击结果 Alice 认为在和 Bob 共享密钥  $K$ ,但事实上是在和 Malice 共享密钥  $K$ 。



1. Alice 发给 Malice(“Trent”):  $Alice, Bob$ ;
2. Malice(“Alice”)发给 Trent:  $Alice, Malice$ ;
3. Trent 找出密钥  $K_{AT}$ 、 $K_{MT}$ , 随机产生  $K_{AM}$  并发给 Alice:  $\{K_{AM}\}_{K_{AT}}, \{K_{AM}\}_{K_{MT}}$ ;
4. Alice 解密  $\{K_{AM}\}_{K_{AT}}$  并发给 Malice(“Bob”):  $Trent, Alice, \{K_{AM}\}_{K_{MT}}$ ;
5. Malice(“Bob”)发给 Alice:  $\{你好 Alice, 我是 Bob!\}_{K_{AM}}$ 。

Malice 开始截获由 Alice 发送给 Trent 的初始消息。这条消息的意思是让 Trent 产生一个由 Alice 与 Bob 共享的会话密钥  $K_{AM}$ 。Malice 通过把 Bob 的身份改为自己的身份来改变这条消息并把它发送给 Trent。Trent 会认为 Alice 想和 Malice 交谈,因此他产生了一个 Malice 和 Alice 共享的新的会话密钥  $K_{AM}$ ,同时 Trent 用和这两个主体共享的长期密钥分别进行加密。由于 Alice 不能区分发送给其他用户的加密消息,所以她不会察觉出这种改变。Malice 随后就可截获 Alice 打算发给 Bob 的信息,而 Bob 并不知道他被要求运行这个协议。攻击的结果是, Alice 认为和 Bob 成功地完成了这个协议,而实际上 Malice 知道  $K_{AM}$ ,因此可以伪装成 Bob 窃听 Alice 发送给 Bob 的所有消息。注意,仅在 Malice 对于 Trent 是一个已知合法用户的情况下,这种攻击才会成功。这又是一个现实的假设——内部攻击者通常比外部攻击者更有威胁。

我们已经看到,上述攻击结果是 Malice 通过改变 Bob 的身份造成的。我们注意到这种改变是可能的,因为 Bob 的身份是用明文发送的。这就提示我们通过隐藏 Bob 的身份来修改协议。

#### 2.6.2.2 一种修补

在理解了上述 Malice 改变 Bob 身份的攻击后,要修改协议“来自 Trent 的会话密钥”看来就很简单了。例如,我们可以对协议进行修改,将第一条消息中 Bob 的身份处理成为秘密的,用 Alice 和 Trent 之间的共享密钥进行加密。也就是说,协议“来自 Trent 的会话密钥”中第一条消息应正确地修改为

1. Alice 发送给 Trent:  $Alice, \{Bob\}_{K_{AT}}$ ;

注意,这里 Alice 的身份仍为明文是必要的,以便 Trent 能知道他应用哪个密钥对密文部分进行解密。

#### 2.6.2.3 另一种攻击

然而,上述修改方法没有为“来自 Trent 的会话密钥”提供一种彻底的修补。例如,不难看出, Malice 会执行:

1. Malice(“Alice”)向 Trent 发送:  $Alice, \{Malice\}_{K_{AT}}$ ;

协议的其余部分与攻击 2.1 相同。如果 Malice 起初不知道 Alice 打算和谁运行这个协议,因为消息中必须包含 Bob 的地址以便网络正确传递信息,于是当他截获到 Alice 发送给 Bob 的消息,就可以知道这部分信息。因此 Malice 最终仍然能成功地伪装成 Bob。注意,在这个攻击中我们假定 Malice 有密文  $\{Malice\}_{K_{AT}}$ ;这可能是因为 Malice 已经从 Alice 和 Malice 先前的一个协议运行中(一个正确的运行)将它记载下来了。

#### 2.6.2.4 再一种攻击

事实上,对协议“来自 Trent 的会话密钥”(或上述它的修补)还有一种攻击方法,它并不依赖改变任何主体的身份。而是, Malice 将 Trent 发给 Alice 的消息(协议“来自 Trent 的会话密钥”第 2 条中的消息)改变为

- Malice(“Trent”)向 Alice 发送:  $\{K'\}_{K_{AT}}, \dots;$



这里  $K'$  是在以前协议运行中(一个正确的运行)Alice 和 Malice 之间的一个会话密钥,这样 Malice 已经记下了密文组  $\{K'\}_{K_{AT}}$ 。这个攻击运行的其余部分和攻击 2.1 中的类似:Malice 截获由 Alice 发给 Bob 的随后的消息,最后以伪装成 Bob 向 Alice 确认:

Malice(“Bob”)发送给 Alice:  $\{\text{你好 Alice, 我是 Bob!}\}_K$ 。

可以通过改变或不改变 Bob 的身份来对修补后的“来自 Trent 的会话密钥”协议进行攻击,这一事实清楚地表明,对协议“来自 Trent 的会话密钥”第一条中的 Bob 的身份进行加密处理也不能使协议安全。迄今列出的攻击表明,Malice 改变协议中的某些消息而不被察觉是可能的。这就提示,协议需要一个能防止消息被篡改的安全服务。

这就为我们带来了下述安全服务。

### 2.6.3 消息认证协议

通过目前所列出的攻击我们已经知道, Malice 总可以改变协议的某些消息而不被发现。的确,迄今为止所设计的协议均不能提供任何密码保护来防止消息篡改。因而,修补这些协议的一个方法就是提供这种保护。这种保护应该能够让有正确密钥的合法主体检测任何受保护协议消息的未授权改变。这样的保护或安全服务称为消息认证(在一些文献中也称为数据完整性,我们将在第 11 章区分这两个概念)。

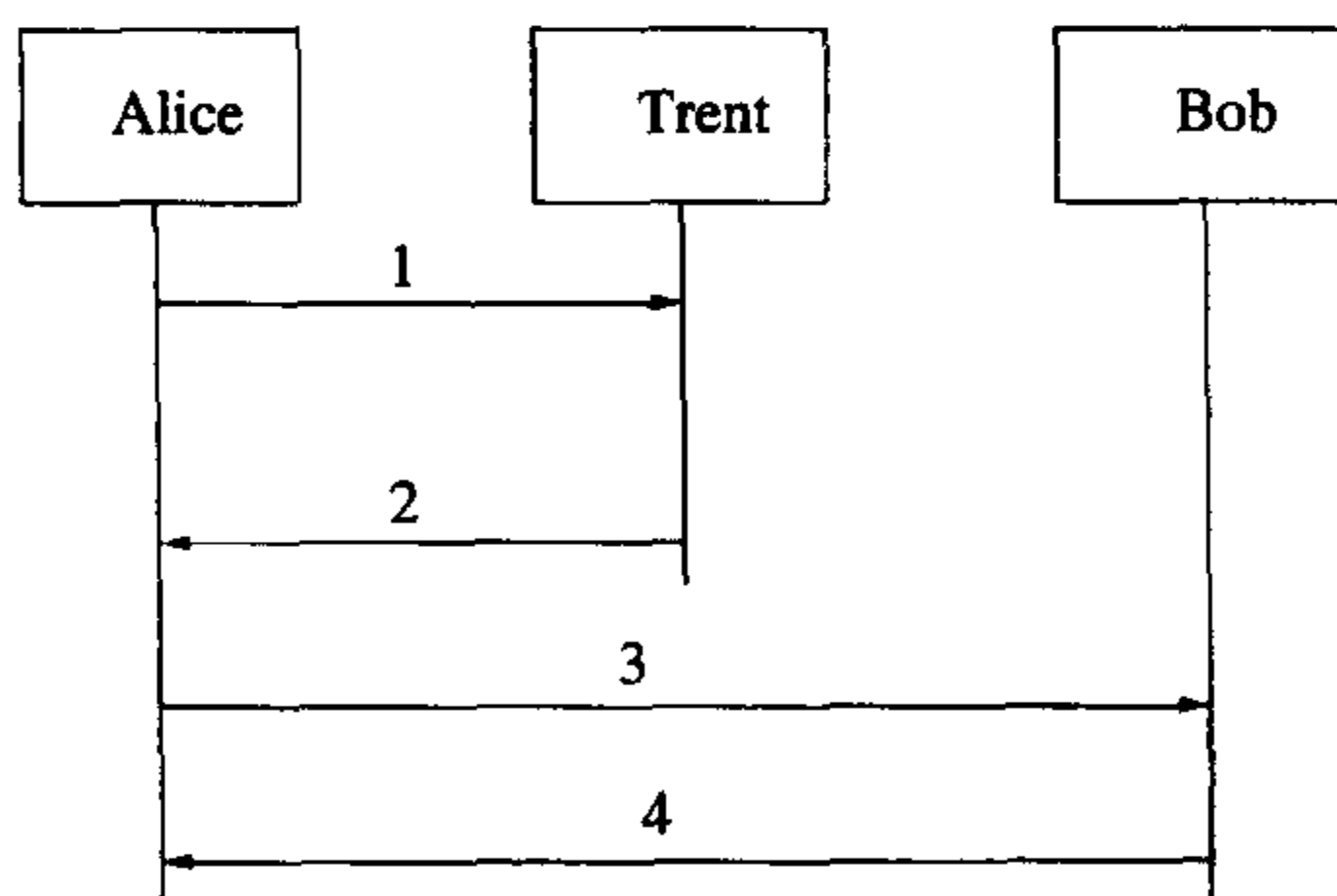
#### 2.6.3.1 “消息认证”协议

我们注意到, Malice 对协议消息的修补会引起以下两种结果:要么在错误的主体之间分享一个会话密钥,要么建立一个错误的会话密钥。因此我们建议,消息认证保护应在要建立的会话密钥和它的使用者之间提供一种密码绑定,这就引出一个新的协议:协议 2.3,其中 Alice 和 Bob 的身份要包含在由 Trent 发送的加密消息部分中。我们称这种新的协议为“消息认证”。

#### 协议 2.3 消息认证

假定 Alice 和 Trent 共享密钥  $K_{AT}$ , Bob 和 Trent 共享密钥  $K_{BT}$ 。

目标 Alice 和 Bob 希望建立一个新的共享秘密密钥  $K$ 。



1. Alice 向 Trent 发送:  $Alice, Bob$ ;
2. Trent 找出密钥  $K_{AT}, K_{BT}$ , 随机产生  $K$  并发送给 Alice:  $\{Bob, K\}_{K_{AT}}, \{Alice, K\}_{K_{BT}}$ ;
3. Alice 解密  $\{Bob, K\}_{K_{AT}}$ , 验证 Bob 的身份, 向 Bob 发送:  $Trent, \{Alice, K\}_{K_{BT}}$ ;
4. Bob 解密  $\{Alice, K\}_{K_{BT}}$ , 验证 Alice 的身份, 向 Alice 发送:  $\{\text{你好 Alice, 我是 Bob!}\}_K$ 。

我们应特别注意“消息认证”协议的说明部分,它指出:

3. Alice(解密  $\{Bob, K\}_{K_{AT}}$ ), 验证 Bob 的身份, ……
4. Bob(解密  $\{Alice, K\}_{K_{BT}}$ ), 验证 Alice 的身份, ……

在这个“消息认证”协议中,检验主体身份的步骤使这个协议和他的前身(协议“来自 Trent 的会话密钥”和它的“修补”等)有了关键性区别。这些检验步骤仅可能在用正确的密钥对相应的密文进行正确解密之后执行。因此,消息认证服务是通过接收者执行的“解密-检验”密码运算实现的,这个消息认证服务使接收者能够验证将要建立的会话密钥同它的使用者之间的密码绑定是否正确。正确的解密结果应该能够表明相应的密文组在传递中没有被改变。这就是“消息认证”协议防止迄今为止所给出的攻击的方法。

我们应该指出,要实现消息认证,“解密-检验”运算(由接收者执行)并不是一个正确的“运算模式”。在第 17 章中我们将看到,正确的运算模式应该是“再加密-检验”(也是由接收者执行)。在这一章我们使用一个不正确或不精确的运算模式的原因是,“发送者加密”和“接收者解密”是我们目前惟一可供使用的密码运算。

因为我们将使用一个不正确的运算模式实现消息认证服务,所以有必要明确说明我们的加密算法必须满足一个额外要求的性质。性质如下[其标号(iii)随 2.2 节中“用符号  $\{M\}_K$  表示的完善加密”的另外两条性质的标号]:

**性质 2.2** 用符号  $\{M\}_K$  表示的完善加密(用于消息认证服务)

- iii) 如果没有密钥  $K$ ,即使知道明文  $M$ ,要想更改  $\{M\}_K$  而不被接收者在解密阶段发现是不可能的。

为了说明这条性质的重要性,下面我们在假设完善加密算法不具有上面消息认证性质(即假定加密算法只具有 2.2 节所列的完善保密性质)的情况下,给出对“消息认证”协议的一种攻击。为了方便说明,我们将协议中的密文组块

$$\{Bob, K\}_{K_{AT}}, \{Alice, K\}_{K_{BT}}$$

修改为下面的表示形式

$$\{Bob\}_{K_{AT}}, \{K\}_{K_{AT}}, \{Alice\}_{K_{BT}}, \{K\}_{K_{BT}}$$

采用这种密文组表示的含义是,虽然已经破坏了主体身份与会话密钥之间的密码绑定关系,但对要加密的任意明文,加密仍然保持完善保密服务。使用这种完善加密方案的“消息认证”协议的 2、3、4 条消息如下:

2. Trent……, 向 Alice 发送:  $\{Bob\}_{K_{AT}}, \{K\}_{K_{AT}}, \{Alice\}_{K_{BT}}, \{K\}_{K_{BT}}$ ;
3. Alice 解密  $\{Bob\}_{K_{AT}}$  和  $\{K\}_{K_{AT}}$ , 检验 Bob 的身份……
4. Bob 解密  $\{Alice\}_{K_{BT}}$  和  $\{K\}_{K_{BT}}$ , 检验 Alice 的身份……

显然,协议不能对主体的身份提供机密性保护。仅仅观察协议中消息在网络间的流动(从发送者到接收者),Mlice 就能精确地确定密文组  $\{Bob\}_{K_{AT}}$  和  $\{Alice\}_{K_{BT}}$  中的明文内容。因此修改后的协议与“来自 Trent 的会话密钥”协议本质上是相同的,也会招致与 2.6.2 节介绍的本质上相同的攻击。作为练习,读者可以实施一下这些攻击。

### 2.6.3.2 对“消息认证”协议的攻击

尽管使用具有消息认证特性的加密算法,“消息认证”协议仍会受到攻击。这个问题出自 Trent 和他的客户之间初始所共享的长期密钥和每次协议运行所产生的会话密钥在质量上的差别。

首先,我们注意到,Trent 和他的每个客户的关系是长期的,这意味在他们之间的共享密钥是长期密钥。一般,在认证服务器和其客户之间建立密钥,要比在两个客户主体之间建立会话密钥更困难,也更昂贵(后者需要彻底的安全检查程序,甚至需要面对面地接触)。所幸的是,这样的密钥主要在认证协议中偶尔用来加密少量具有很小冗余度的消息,所以这种密钥的使用几乎不会为密码分析提供信息。因此,认证服务器与其客户之间的共享密钥可以长期使用,我们常称之为长期密钥。

另一方面,我们回忆 2.5 节讨论过的密钥管理原则,它强调一个会话密钥只能在一次会话中使用。因而,会话密钥建立协议的任何两次运行所产生的会话密钥都不应是相同的。但“消息认证”协议并非如此。协议的攻击运行方式会破坏会话密钥管理原则。在这种攻击中,Mlice 首先要做的就是截获 Alice 的请求(见协议 2.3)。

1. Alice 向 Mlice(“Trent”)发送:……

然后插入消息行 2 如下:

2. Malice(“Trent”)向 Alice 发送: $\{Bob, K'\}_{K_{AT}}, \{Alice, K'\}_{K_{BT}}$

这里,两个包括  $K'$  的密文组是旧消息的重放,旧消息是 Malice 从以前协议(Alice 与 Bob 之间的一次正常运行)的运行中记录的。因此这个攻击会使 Alice 与 Bob 重新使用那个他们本不该使用的旧会话密钥  $K'$ 。注意,既然是旧的,Malice 很可能已经发现了该密钥(也许是因为粗心的主体丢掉了,或者是由于我们在 2.5 节已经讨论过的会话密钥的其他弱点),于是他可以窃听 Alice 与 Bob 之间的秘密会话通信,或者模仿 Bob 与 Alice 交谈。

以上形式的攻击称之为消息重放攻击。

### 2.6.4 询问-应答协议

用户可使用几种不同的机制来检验协议中的消息是否为旧消息的一次重放。这些机制将在第 11 章详细讨论。但是现在,我们要利用一个叫询问-应答(也叫握手)的著名方法来改进我们的协议。利用这种方法,在协议一开始,Alice 生成一个新的随机数  $N_A$ ,并把它连同新的会话密钥请求一起发送给 Trent。如果返回一个相同的值( $N_A$ )和一个会话密钥,且这两个值是通过密码技术绑定在一起的,而且这个绑定提供一种消息认证服务(也就是说,Alice 可以验证包含  $N_A$  密文的消息完整性),那么在收到 Alice 的随机数  $N_A$  后,Alice 就可以推断出 Trent 所生成的密码绑定。另外,回想一下我们关于 Trent 信任的规定(见 2.4 节);Alice 知道 Trent 始终忠实地遵守协议。所以在 Trent 收到 Alice 的随机询问后,他确实生成了一个新的会话密钥。因此会话密钥应该是新的(或新鲜的、当前的),即不是旧密钥的一次重放。由 Alice 生成并使询问-应答机制得以运行的随机数称为 **nonce**,表示仅使用一次的数字[62]。

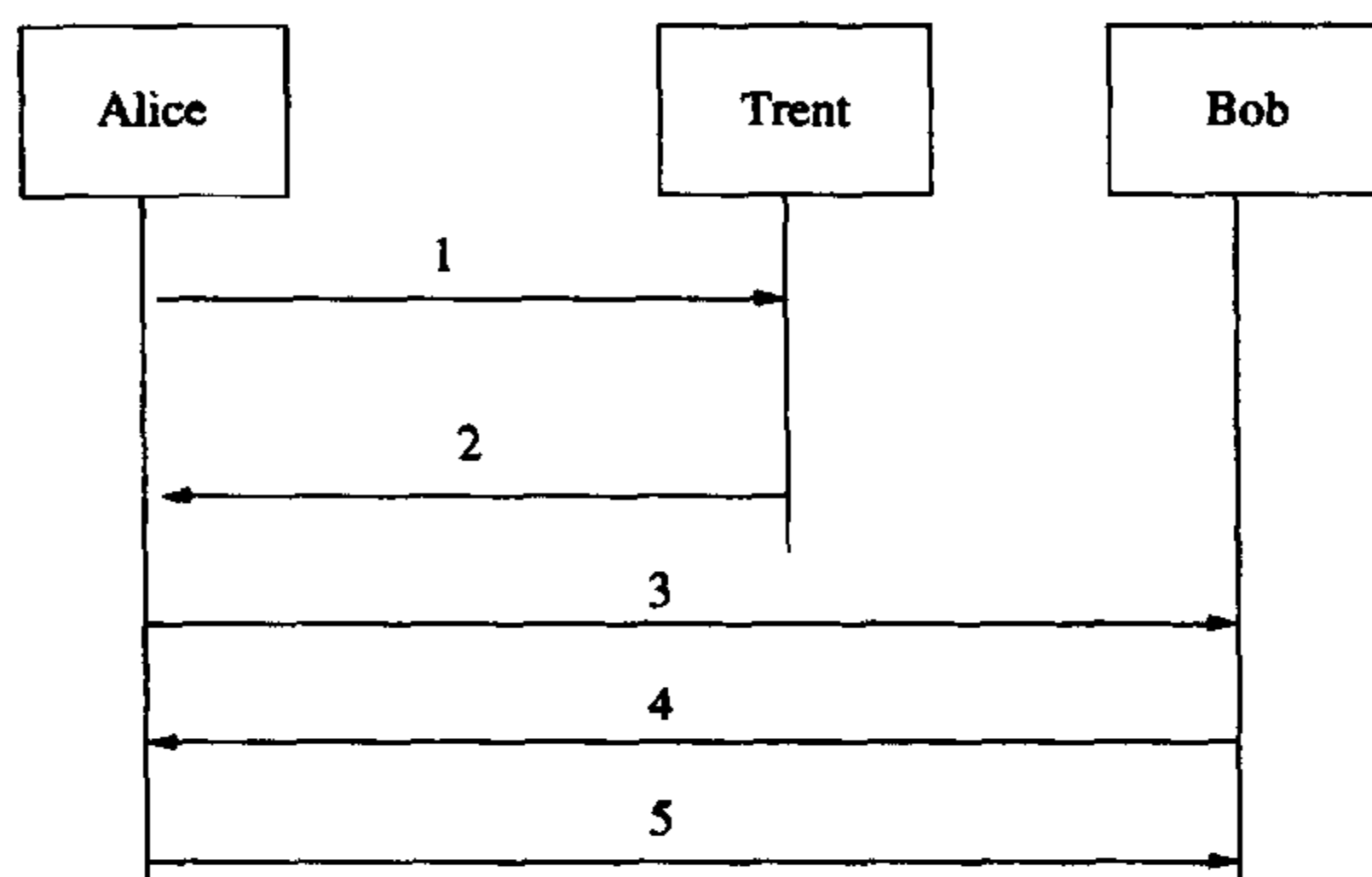
#### 2.6.4.1 “询问-应答”协议(Needham-Schroeder)

协议 2.4 详细说明了一个新的协议,这个协议利用询问-应答机制使 Alice 能够检查会话密钥的新鲜性。我们暂时称它为“询问-应答”(不久我们将换一种叫法)。

##### 协议 2.4 询问-应答协议

假定 Alice 和 Trent 共享密钥  $K_{AT}$ ; Bob 和 Trent 共享密钥  $K_{BT}$ 。

目标 Alice 和 Bob 希望建立一个新的共享密钥  $K$ 。



1. Alice 随机生成  $N_A$ , 向 Trent 发送:  $Alice, Bob, N_A$ ;
2. Trent 随机生成  $K$ , 向 Alice 发送:  $\{N_A, K, Bob, \{K, Alice\}_{K_{BT}}\}_{K_{AT}}$ ;
3. Alice 解密  $N_A$ 、验证她的 nonce, 验证 Bob 的身份并发给 Bob:  $Trent, \{K, Alice\}_{K_{BT}}$ ;
4. Bob 解密、验证 Alice 的身份, 生成随机数  $N_B$ , 向 Alice 发送:  $\{我是 Bob! N_B\}_K$ ;
5. Alice 发向 Bob 发送:  $\{我是 Alice! N_B - 1\}_K$ 。

在“询问-应答”协议中, Bob 也生成一个 nonce( $N_B$ ), 但这个 nonce 不发送给 Trent, 因为在这个协议中, Bob 不直接与 Trent 联系。Bob 的 nonce 发送给 Alice, 然后 Alice 对其稍做修改(如减 1), 并对 Bob 应答。所以, 如果 Alice 对会话密钥  $K$  的新鲜性满意, 并用它对 Bob 新产生的 nonce 做应答, 那么 Bob 就推断出会话密钥是新鲜的, 对会话密钥的相互信赖就这样建立起来了。

经过一系列的步骤之后, 我们已经轮到讨论“询问-应答”协议, 它可能是认证和密钥建立协议中最著名的协议。它正是 Needham 和 Schroeder 协议, 两人于 1978 年发表[215]。以后我们称这个协议为 Needham-Schroeder 对称密钥认证协议。这一协议也是整个有关协议类的基础。

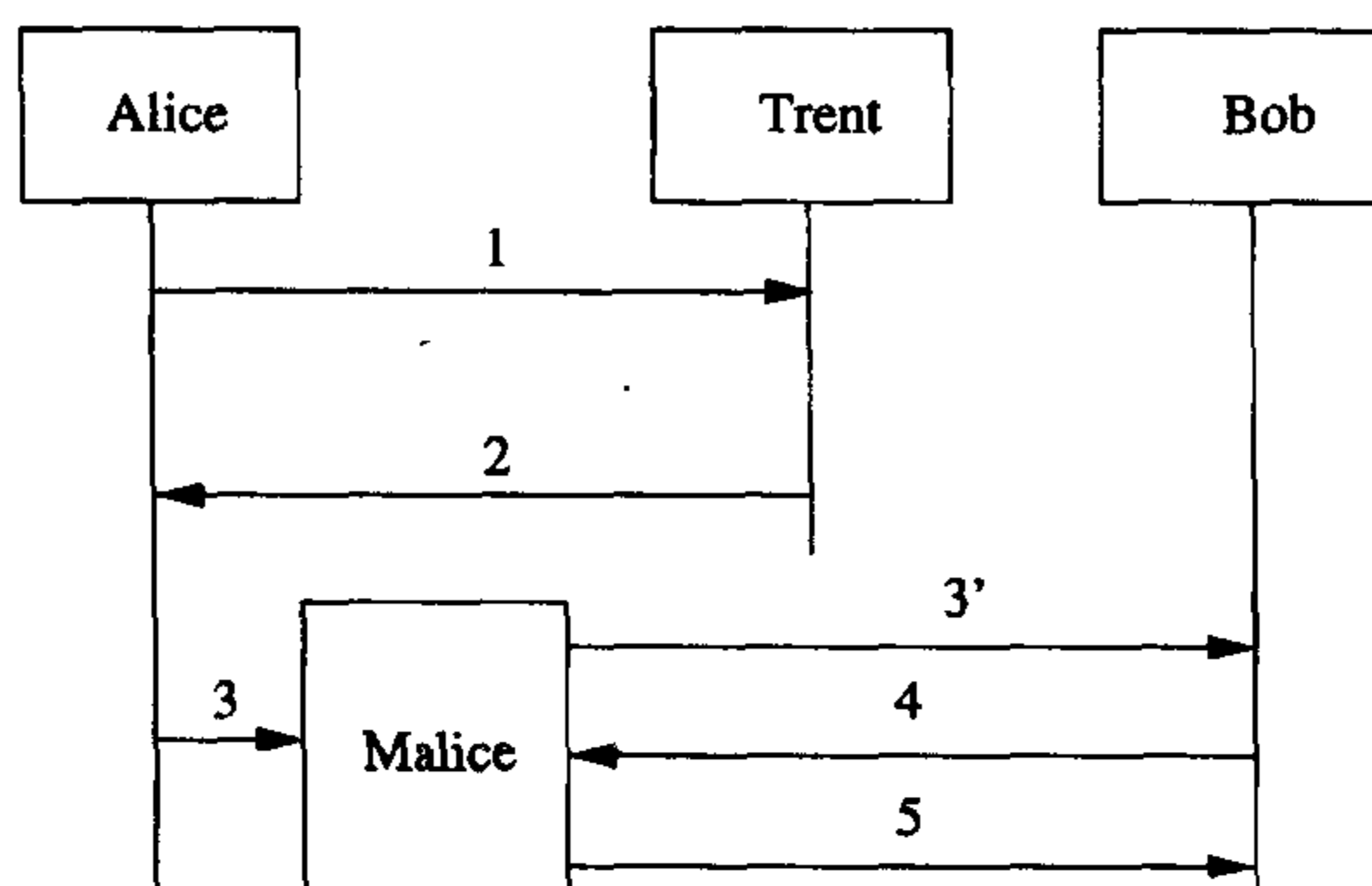
#### 2.6.4.2 对 Needham-Schroeder 对称密钥认证协议的攻击

遗憾的是, Needham-Schroeder 协议对应于 1981 年 Denning 和 Sacco 所发现的攻击是脆弱的。在 Denning 和 Sacco 攻击中, Malice 截获了 Alice 发送和接收的消息行 3、4、5, 并用自己的消息替代这些消息, 攻击 2.2 中给出了这个攻击。

##### 攻击 2.2 对 Needham-Schroeder 对称密钥认证协议的攻击

攻击结果

Bob 认为他正在和 Alice 共享一个新的会话密钥,但实际上这个密钥并不是新的, Malice 有可能知道该密钥。



1 和 2.(与正常运行中相同)

3. Alice 向 Malice(“Bob”)发送:……

3'. Malice(“Alice”)向 Bob 发送:  $\{K', Alice\}_{K_{BT}}$ ;

4. Bob 解密并验证 Alice 的身份,向 Malice(“Alice”)发送:  $\{我是 Bob! N_B\}_K$ ;

5. Malice(“Alice”)向 Bob 发送:  $\{我是 Alice! N_B - 1\}_K$ 。

在攻击中, Malice 从消息行 3 开始主动拦截 Alice 发给 Bob 的消息。然后他完全阻截 Alice 的通信信道,并重放旧的会话密钥数据  $\{K', Alice\}_{K_{BT}}$ ,这是他对以前 Alice 和 Bob 之间的运行协议所做的记录。根据我们关于旧会话密钥脆弱性的假设, Malice 可能知道  $K'$  的值,因此他可以伪装成 Alice,发起对 Alice 与 Bob 之间谈话的攻击。

应该指出,老会话密钥的脆弱性只是这种攻击危险性的一个方面。这种攻击的另一个危险是, Malice 能成功地破坏认证所要达到的一个重要目标。我们将在 11.2.2 节中具体说明这个认证目标,并在 11.7.1 节去看 Malice 如何成功地破坏这个目标。

### 2.6.5 实体认证协议

在 Needham-Schroeder 协议(Alice 和 Trent 之间的交互部分)中,询问-应答机制提供了一种称为**实体认证**的安全服务。类似于消息认证,实体认证服务也是(由验证主体)通过验证一种密码算法获得的。这两种服务的不同之处在于后者提供了一种主体(示证主体)活现性的证据。如果示证主体在最近的一个事件后执行了一种密码运算,而且验证主体知道这是个最近的事件,那么就证明了证据的活现性。在 Needham-Schroeder 协议中,当 Alice 收到消息行 2 时, Alice 通过解密运算揭示她的 nonce  $N_A$ ,表明在她发送 nonce  $N_A$  之后, Trent 仅进行了加密运算(因为所使用的密钥是她和 Trent 共享的)。因此在该事件之后, Alice 知道 Trent 是真实存在着的。这就完成了一个从 Trent 到 Alice 的实体认证。

但对于 Bob 来说,在 Needham-Schroeder 协议中,他没有关于 Trent 活现性的实体认证证据。

像通常一样,一旦点出了问题,就不难提出对它的修补办法: Trent 应该在实体认证中向两个客户实体证实自己。例如,这可以通过 Bob 也向 Trent 发送一个 nonce 来实现,这个 nonce 也应该包含在 Trent 返回的会话密钥消息中。这种修补方法将增加协议中的消息流(增加一个 Bob 和 Trent 间的握手)。Denning 和 Sacco 建议使用**时戳(timestamp)**来避免增加消息流[95]。

### 2.6.5.1 时戳

设  $T$  表示一个时戳, Denning 和 Sacco 所建议的修补法如下:

1. Alice 发送给 Trent:  $Alice, Bob$ ;
  2. Trent 发送给 Alice:  $\{Bob, K, T, \{Alice, K, T\}_{K_{BT}}\}_{K_{AT}}$ ;
  3. Alice 发送给 Bob:  $\{Alice, K, T\}_{K_{BT}}$ ;
  4. ....
  5. ....
- (4 和 5 与 Needham-Schroeder 协议中的 4 和 5 相同)。

当 Alice 和 Bob 收到 Trent 所发出的协议消息时, 通过验证

$$|Clock - T| < \Delta t_1 + \Delta t_2$$

他们就能够证明消息不是重放的。这里  $Clock$  是接收者的当地时间,  $\Delta t_1$  是 Trent 的时钟和当地时钟之间正常差值的区间,  $\Delta t_2$  是所期望的网络时延区间。如果每一个客户主体都能参考一个标准时钟源来人工地设置自己的时钟,  $\Delta t_1$  的取值大约为一两分钟就足够了。只要  $\Delta t_1 + \Delta t_2$  的值比自上次使用这个协议以来的时间间隔值小, 这种方法就能抗攻击 2.2 的重放攻击。由于时戳  $T$  是在密钥  $K_{AT}$  和  $K_{BT}$  下加密的, 如果加密方案是完善的, 要模仿 Trent 是不可能的。

Needham 和 Schroeder 已经考虑了时戳的用处, 但是他们排除了使用时戳方案, 因为这需要一个能普遍可以获取的高质量时钟值[214]。

### 2.6.6 一个使用公钥密码体制的协议

这一章要介绍的最后一个协议称之为 Needham-Schroeder 公钥认证协议[215]。我们在这里介绍这个协议有两个理由, 这两个理由都纳入到了本章的讨论中。首先, 这一协议能使我们初步熟悉公钥密码体制的使用。其次, 我们将给出对这个协议的一种巧妙攻击法。这个协议虽然看似简单, 但却是在协议公布十七年后才发现这种攻击。

#### 2.6.6.1 公钥密码体制

我们用密钥  $K_A$  来表示 Alice 的公钥, 用  $K_A^{-1}$  表示相应的私钥 (Alice 的私钥)。假定只有 Alice 拥有她的私钥。密文组

$$\{M\}_{K_A}$$

表示使用 Alice 的公钥  $K_A$  对明文  $M$  的完善加密。假定对以上密文解密必须使用相应的私钥  $K_A^{-1}$ 。由于假定只有 Alice 拥有私钥, 所以只有她才能解密恢复出明文  $M$ 。类似地, 密文组

$$\{M\}_{K_A^{-1}}$$

表示使用 Alice 的私钥  $K_A^{-1}$  对明文  $M$  的完善加密, 只有用 Alice 的公钥  $K_A$  才能解密。知道  $K_A$  是 Alice 的公钥, 用  $K_A$  解密的过程使我们知道密文  $\{M\}_{K_A^{-1}}$  是由 Alice 生成的, 这是因为生成  $\{M\}_{K_A^{-1}}$  需要一个密钥, 而这个密钥只有 Alice 拥有。因此, 密文组  $\{M\}_{K_A^{-1}}$  也称为 Alice 对消息  $M$  的(数字)签名, 利用  $K_A$  的解密过程称为关于 Alice 对消息  $M$  签名的验证。



### 2.6.6.2 Needham-Schroeder 公钥认证协议

假设 Trent 拥有他所服务的所有客户主体的公钥,每一个客户主体也有一个已认证 Trent 公钥的副本。Needham-Schroeder 公钥认证协议由协议 2.5 具体说明。

#### 协议 2.5 Needham-Schroeder 公钥认证协议

假定 Alice 的公钥是  $K_A$ , Bob 的公钥是  $K_B$ , Trent 的公钥是  $K_T$ 。

目标 Alice 和 Bob 建立一个新的共享秘密。

1. Alice 发给 Trent :  $Alice, Bob$ ;
2. Trent 发给 Alice:  $\{K_B, Bob\}_{K_T^{-1}}$ ;
3. Alice 验证 Trent 对“ $K_B, Bob$ ”的签字,随机生成她的 nonce  $N_A$  并发给 Bob:  $\{N_A, Alice\}_{K_B}$ ;
4. Bob 解密、验证 Alice 的身份并发给 Trent:  $Bob, Alice$ ;
5. Trent 发给 Bob:  $\{K_A, Alice\}_{K_T^{-1}}$ ;
6. Bob 验证 Trent 对“ $K_A, Alice$ ”的签字,随机生成他的 nonce  $N_B$  并发给 Alice:  $\{N_A, N_B\}_{K_A}$ ;
7. Alice 解密,并发给 Bob:  $\{N_B\}_{K_B}$ 。

这里 Alice 是一个发起者,他在 Trent 的帮助下,寻求与响应者 Bob 建立会话。第 1 步, Alice 向 Trent 发一个消息,请求 Bob 的公钥。第 2 步, Trent 用自己的私钥  $K_T^{-1}$  对  $K_B$  和 Bob 的身份加密并回送给 Alice 应答(为了防止 2.6.2 节中的一类攻击)。这就形成了 Trent 对协议消息的数字签字,它使 Alice 相信第二步中的消息来自 Trent (Alice 应该用 Trent 的公钥验证这个签字)。之后, Alice 通过随机选择一个 nonce  $N_A$ ,并用 Bob 的公钥将  $N_A$  和自己的身份加密后发送给 Bob(第 3 步)来寻求和 Bob 建立起连接。Bob 收到这个消息后,对这个消息解密获得 nonce  $N_A$ ,他请求(第 4 步)并收到(第 5 步) Alice 的公钥认证副本。然后,他用 Alice 的公钥对  $N_A$  和自己新产生的 nonce  $N_B$  加密,送给 Alice(第 6 步)。当 Alice 收到消息后,她应该确信她正和 Bob 交谈,这是因为仅有 Bob 能解密消息 3 并获得  $N_A$ ,而且这一项必定是在她发送 nonce 之后才进行(一个最近的行动)。接着, Alice 用 Bob 的公钥加密  $N_B$  后回送给 Bob。当 Bob 收到消息后,他也会确信他正和 Alice 交谈,这是因为只有 Alice 能解密消息 6 来获得 nonce  $N_B$ (也是一个最近的行动)。这样,成功运行这个协议就实现了共享 nonce  $N_A$  和  $N_B$  的建立,它们就是只由 Alice 和 Bob 共享的秘密。还要注意,由于共享的这些秘密都是两方主体最近的贡献,所以具有新鲜性。而且,只要每一个主体自己所贡献的部分是足够随机的,他就应该相信这个秘密的随机性。

Needham 和 Schroeder 建议,从大的空间来选  $N_A$  和  $N_B$ ,就可以为 Alice 和 Bob 接下来的安全通信初始化一个共享密钥(“作为顺序排列加密数据组的基础”)[215]。

Denning 和 Sacco 曾指出,这个协议并不能保证客户主体所获得的公钥是新的而不是旧的重放,它可能是已泄露的密钥[95]。这个问题可以用多种方法来解决,例如在密钥传送中加入时戳<sup>①</sup>。下面我们假定从 Trent 获得的客户公钥是新的并且是好的。

<sup>①</sup> Denning 和 Sacco 提出这样一种修补[95],但是由于另一个原因他们的修补也存在缺陷。我们将在 11.7.7 节看到他们的修补并研究存在缺陷的原因。

### 2.6.6.3 对 Needham-Schroeder 公钥认证协议的攻击

Lowe 发现了对 Needham-Schroeder 公钥认证协议的一种攻击[182]。

Lowe 观察到,这个协议可以看做是两个逻辑分离协议的交织。第 1、2、4、5 步关系到如何获得公钥,而第 3、6、7 步关系到 Alice 和 Bob 的认证。因此,可以假定每个主体在开始时都有其他主体的公钥的认证副本。我们的注意力只限于下面的步骤(这里仅列出了消息流,读者可具体参考协议 2.5):

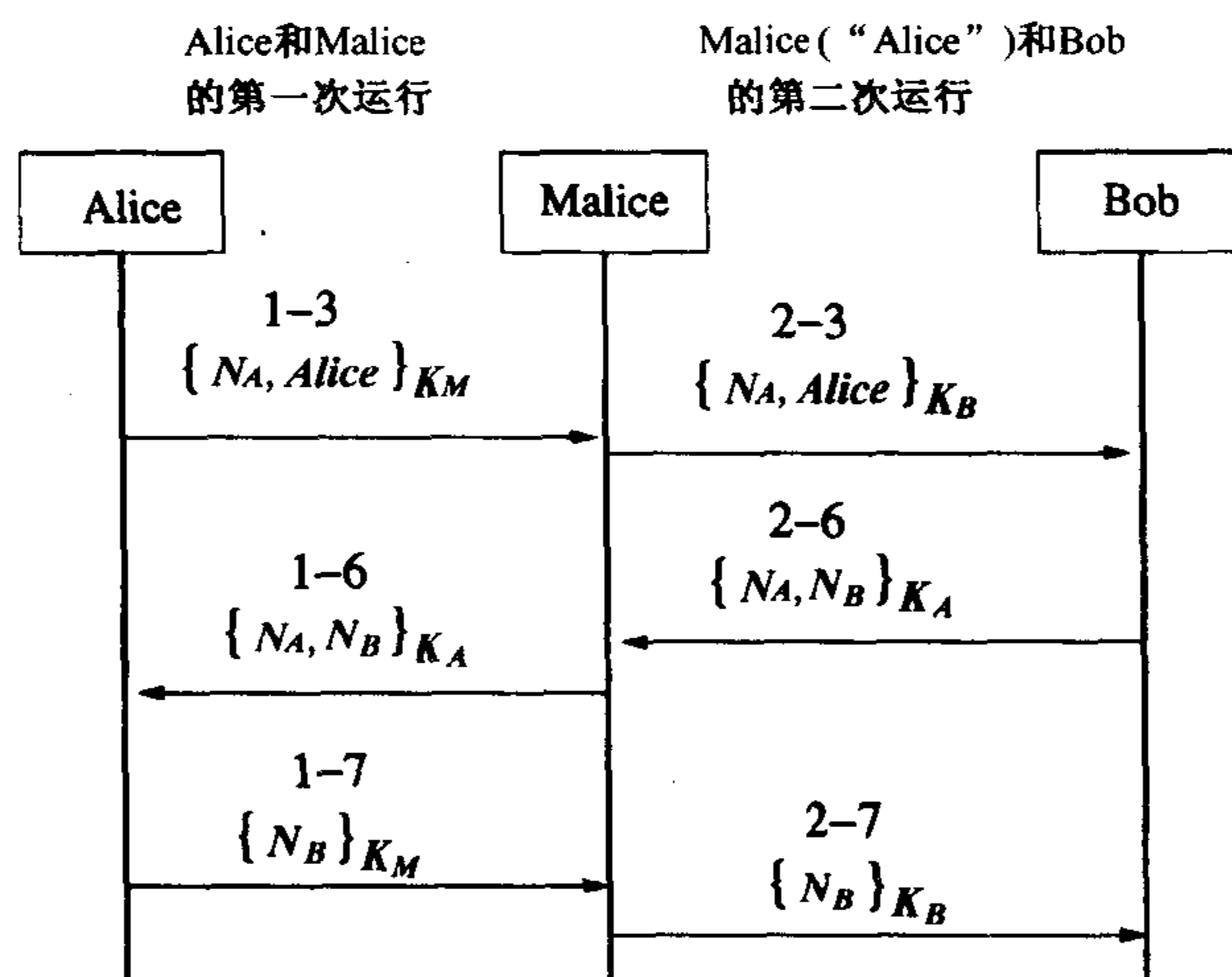
3. Alice 向 Bob 发送:  $\{N_A, Alice\}_{K_B}$ ;
6. Bob 向 Alice 发送:  $\{N_A, N_B\}_{K_A}$ ;
7. Alice 向 Bob 发送:  $\{N_B\}_{K_B}$ 。

我们将分析 Malice 是如何能和这个协议交互的。假设在这个系统中 Malice 是一个合法主体,因此其他主体可能要和 Malice 建立标准会话。确实,下面的攻击就是始于 Alice 试图和 Malice 建立一个会话。攻击 2.3 描述了这个攻击。

#### 攻击 2.3 Lowe 的关于 Needham-Schroeder 公钥认证协议的攻击

假定 Alice 的公钥是  $K_A$ , Bob 的公钥是  $K_B$ , Malice 的公钥是  $K_M$ 。

攻击结果 Bob 认为他正和 Alice 共享秘密  $N_A, N_B$ ,但实际上正和 Malice 共享。



这个攻击包括两次同时运行该协议。在第一次运行中(第 1-3 步、1-6 步和 1-7 步), Alice 和 Malice 建立了一个合法的会话。在第二次运行中(第 2-3 步、2-6 步和 2-7 步), Malice 伪装成 Alice 同 Bob 建立了一个假的会话。在第 1-3 步中, Alice 开始和 Malice 建立一个正常的会话,并向他发送一个 nonce  $N_A$ 。在第 2-3 步中, Malice 假冒 Alice 试图同 Bob 建立一个假会话,将来自 Alice 的  $N_A$  发给 Bob。Bob 在第 2-6 步中选择一个新的 nonce  $N_B$  应答,并试图把  $N_B$  连同  $N_A$  回送给 Alice。Malice 截获这条消息,但不能对它解密,因为它是用 Alice 的公钥加密的消息。因此, Malice 在第 1-6 步中通过把该消息发送给 Alice, 寻求利用 Alice 为他解密。注意,这个消息正是 Alice 在协议第一次运行所期望的形式。Alice 对这个消息解密获得  $N_B$ , 并在 1-7 步回送

给 Malice(用 Malice 的公钥加密),接着 Malice 可以解密来获得,并在 2-7 步回送给 Bob。这样就完成了协议的第二次运行。因此 Bob 相信 Alice 已和他正确地建立了一个会话,并且只由他们俩共享此秘密 nonce  $N_A$  和  $N_B$ 。

在这种攻击中, Malice 成功的关键步骤是 Alice 无意地为他解密了 Bob 的 nonce  $N_B$ 。我们说当一个主体无意地为攻击者执行了一个密码运算时,该主体就被用做预言机(oracle)或提供了预言机服务(oracle service)。在这本书中我们将会看到许多预言机服务的例子,我们会逐步将其发展成一种方法学。密码算法和协议应该如此地设计:即使用户为攻击者提供预言机服务,它们也是安全的。

我们可以想像这种攻击的如下结果。Malice 可以在随后的提出会话密钥的消息中加入共享的 nonce, Bob 将会相信这条消息源于 Alice。同样,如果 Bob 是一家银行, Malice 就可能伪装成 Alice 发出这样一条消息:

Malice(“Alice”)向 Bob 发送

$\{N_A, N_B, \text{从我的账户划拨 1 000 000 英镑到 Malice 的账户}\}_{K_B}$ 。

#### 2.6.6.4 一个修补

很容易修补这个协议来防止上述攻击。如果我们在协议消息 6 中加入应答者的身份

6. Bob 向 Alice 发送:  $\{Bob, N_A, N_B\}_{K_A}$ ;

攻击步骤 2-6 将变成

2-6. Bob 向 Malice(“Alice”)发送:  $\{Bob, N_A, N_B\}_{K_A}$ 。

因为现在 Alice 正渴望得到一条包含 Malice 身份的消息, Malice 不能有意地利用 Alice 作为一个解密预言机来成功地在步骤 1-6 中重放这个消息。

这个修补可作为 Abadi 和 Needham 提出的密码协议设计原则的一个例子[1]:

如果主体的身份对于消息的意义来说是必要的,那么为了保险起见,应当在消息中明确置入主体的名字。

然而,我们应当克制住不要宣称这种“修补”方法能得到一个安全的协议。在 17.2.1 节我们披露该协议中的其他几个问题,这些问题是由一个不希望的、称为“解密-检验消息认证”的设计特征引起的(我们已将它标为一个错误的运行模式,见 2.6.3.1 节)。这种设计特征一般都出现在利用公钥或私钥密码技术的认证协议中,并在本章的所有协议中都出现过(在我们对 Needham-Schroeder 公钥认证协议的“修补”中保留了这种设计特征,因此我们的“修补”仍是不正确的)。对 Needham-Schroeder 认证协议(对称密钥和公钥)从方法上的“修补”将在 17.2.3 节给出。

认证协议容易出错的特性,激励人们去考虑系统化方法来开发正确的协议。这一论题将在第 17 章介绍。

## 2.7 本章小结

一些人设计保护系统,另一些人想破坏这些系统。这就是生活的现实,没什么特别的。然而,在本章中我们目睹了认证协议现实中的阴暗部分:作为保护机制,这些协议很容易遭到破坏。

事实上,所有复杂系统都很容易含有设计错误。但是,与提供安全服务的系统不同,其他复杂系统中的用户与环境一般不是敌对的,甚至是友好的。例如,为了避免有问题软件系统崩溃,细心的使用者可能会去学着如何避免某些用法。然而,对于一个信息安全系统,它的环境与它的一些用户总是敌对的:它们之所以存在就是为了攻击系统。对它们利用设计错误当然就是一个求之不得的锦囊袋。

我们已经通过认证协议来说明安全系统是容易出错的。虽然协议最容易出错的原因似乎是由于通信本身的问题,但我们使用认证协议的真正原因,是它们所需的密码技术相对简单,因此更适用于本书开始阶段的引论性目的。我们应当记住,要经常以环境对所有安全系统的敌意来提醒我们,在开发安全系统时应当格外小心。

在后面的几章中,我们将回过头来继续研究认证协议。这种更深入的研究包括:认证协议的原则、结构以及对认证协议攻击的分类方法(第 11 章),现实应用的几个协议的研究案例(第 12 章),开发正确认证协议的形式化方法(第 17 章)。

## 习题

- 2.1 主动攻击者能做什么事情?
- 2.2 在 Dolev-Yao 威胁模型下, Malice 的功能很强,因为他控制着整个开放式通信网,他能在不用正确密钥的情况下解密或加密消息吗? 他能由密文求出加密密钥吗? 他能预测随机值吗?
- 2.3 在认证密钥建立协议中, Trent 的作用是什么?
- 2.4 什么是长期密钥、密钥加密密钥、短期密钥以及会话密钥?
- 2.5 为什么用完善加密和完善消息认证服务后,认证协议仍能被破坏?
- 2.6 什么是 nonce? 什么是时戳? 它们在认证或认证密钥建立协议中的作用是什么?
- 2.7 为什么一些消息在认证或认证密钥建立协议中必须是新鲜的?
- 2.8 主体怎样决定协议消息的新鲜程度?
- 2.9 对于完善加密记号  $\{M\}_K$ , 区分以下三个性质: (i) 消息机密性, (ii) 密钥保密性, (iii) 消息认证。
- 2.10 对“来自 Trent 的会话密钥”协议(协议 2.2)给出另一个攻击,允许 Malice 不仅可以像攻击 2.1 那样,对 Alice 伪装成 Bob,而且同时也可对 Bob 伪装成 Alice,因而使得 Malice 能够转播 Alice 和 Bob 之间的“秘密”通信。  
提示:在 Malice(“Alice”)和 Bob 之间运行攻击 2.1 中的另一个实例。
- 2.11 消息认证和实体认证的区别是什么?
- 2.12 给出对 Needham-Schroeder 认证协议的另一个攻击,在这个攻击中 Alice(和 Trent)是完全离线的。
- 2.13 在 Needham-Schroeder 公钥认证协议中,数字签名是否起着重要作用?  
提示:考虑可将该协议简化为仅包括消息行 2、6 和 7 的形式。

## 第二部分 数学基础

这一部分收集了一些数学资料,给出了基本概念、方法、代数运算的基础、算法过程的基本模块,以及在本书其余部分出现的有关建模、说明、分析、变换和解决不同问题的参考。

该部分包括四章:概率论和信息论(第3章)、计算复杂度(第4章)、代数基础(第5章)和数论(第6章)。这部分内容是一个自足的数学参考指南。在本书的其余章节中,我们所遇到的重要数学问题,都可以在这四章的确切位置上找到该问题的支持事实或基础。因此,本书中我们选用数学材料的方式有助于读者以主动、交互的方式学习现代密码学的数学基础。

对于那些在现代密码学的理论基础和实际应用中非常重要的算法和定理,我们将予以特别注意,并对其进行足够详细的说明。如果我们认为某个定理的证明能够帮助读者提高与学习本书中密码学主题相关的技巧,我们将给出该定理的证明。有时,我们的数学主题发展到不得不利用来自于其他数学分支的事实(如线性代数),而这些内容与本书所讲的密码学技巧没有直接的关系;在这种情况下,我们将简单地使用所需的事实,而不给出证明。

# 标准符号

下面的这些标准符号通用于本书的其余部分。一些符号将在其第一次使用时给出定义，另一些符号将不再进一步说明。

$\emptyset$	空集
$S \cup T$	集合 $S$ 和 $T$ 的并
$S \cap T$	集合 $S$ 和 $T$ 的交
$S \setminus T$	集合 $S$ 和 $T$ 的差
$S \subseteq T$	$S$ 是 $T$ 的子集
$\#S$	集合 $S$ 中的元素数目(比如, $\#\emptyset = 0$ )
$x \in S, x \notin S$	元素 $x$ 属于(不属于)集合 $S$
$x \in_v S$	均匀随机地在集合 $S$ 中选取元素 $x$
$x \in (a, b), x \in [a, b]$	$x$ 属于开区间 $(a, b)$ ( $x$ 属于闭区间 $[a, b]$ )
$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$	自然数集、整数集、有理数集、实数集和复数集
$\mathbb{Z}_n$	模 $n$ 的整数
$\mathbb{Z}_n^*$	模 $n$ 的整数乘法群
$\mathbb{F}_q$	$q$ 个元素的有限域
$\text{desc}(A)$	代数结构 $A$ 的描述
$x \leftarrow D$	根据分布 $D$ 进行赋值
$x \leftarrow_v S$	按 $S$ 为均匀分布进行赋值
$a \pmod b$	模运算: $a$ 被 $b$ 除所得的余数
$x \mid y, x \nmid y$	整数 $y$ 可被(不可被)整数 $x$ 整除
$\stackrel{\text{def}}{=}$	定义为
$\forall$	对所有的
$\exists$	存在
$\text{gcd}(x, y)$	$x$ 和 $y$ 的最大公因子
$\text{lcm}(x, y)$	$x$ 和 $y$ 的最小公倍数
$\log_b x$	$x$ 以 $b$ 为底的对数; $b$ 省略则表示自然对数
$\lfloor x \rfloor$	小于或等于 $x$ 的最大整数
$\lceil x \rceil$	大于或等于 $x$ 的最小整数
$ x $	整数 $x$ 的长度 ( $= 1 + \lfloor \log_2 x \rfloor, x \geq 1$ ), 也表示 $x$ 的绝对值
$\phi(n)$	$n$ 的 Euler 函数
$\lambda(n)$	$n$ 的 Carmichael 函数
$\text{ord}(x)$	群元素的阶
$\text{ord}_n(x)$	(模 $n$ 下) $x$ 的阶



$\langle g \rangle$	由 $g$ 生成的循环群
$\left(\frac{x}{y}\right)$	整数 $x$ 模以整数 $y$ 的 Legendre-Jacobi 符号
$J_n(1)$	$\{x \mid x \in \mathbb{Z}_n^*, \left(\frac{x}{n}\right) = 1\}$
$QR_n$	模整数 $n$ 的二次剩余集合
$QNR_n$	模整数 $n$ 的二次非剩余集合
$\deg(P)$	多项式 $P$ 的次数
$\sum_{i=1}^n v_i, \sum_{i \in S} v_i$	对所有的 $i = 1, 2, \dots, n$ 或者 $i \in S$ , 对数值 $v_i$ 求和
$\prod_{i=1}^n v_i, \prod_{i \in S} v_i$	对所有的 $i = 1, 2, \dots, n$ 或者 $i \in S$ , 对数值 $v_i$ 求积
$\overline{E}$	事件 $E$ 的补事件
$E \cup F$	事件 $E$ 、 $F$ 的和事件, 即或者事件 $E$ 发生, 或者事件 $F$ 发生
$E \cap F$	事件 $E$ 、 $F$ 的积事件, 即事件 $E$ 和事件 $F$ 都发生
$E \subseteq F$	事件 $F$ 包含事件 $E$ , 即事件 $E$ 的发生蕴含事件 $F$ 的发生
$E \setminus F$	事件 $E$ 、 $F$ 的差 ( $= E \cap \overline{F}$ )
$\bigcup_{i=1}^n E_i, \bigcup_{i \in S} E_i$	对所有的 $i = 1, 2, \dots, n$ 或者 $i \in S$ , 对事件 $E_i$ 求和
$\bigcap_{i=1}^n E_i, \bigcap_{i \in S} E_i$	对所有的 $i = 1, 2, \dots, n$ 或者 $i \in S$ , 对事件 $E_i$ 求积
$\text{Prob}[E]$	事件 $E$ 发生的概率
$\text{Prob}[E F]$	已知事件 $F$ 发生的条件下, 事件 $E$ 发生的条件概率
$n!$	$n$ 的阶乘 ( $= n(n-1)(n-2)\cdots 1, 0! = 1$ )
$\binom{n}{k}$	从 $n$ 个中取 $k$ 个的方法数 ( $= \frac{n!}{k!(n-k)!}$ )
$b(k; n, p)$	成功概率为 $p$ 的 $n$ 次贝努利试验中, 有 $k$ 次成功的二项式分布
$O(f(n))$	对某常量 $c > 0$ 和所有足够大的 $n$ , 满足 $ g(n)  \leq c f(n) $ 的函数 $g(n)$
$O_B()$	位运算模式下的 $O()$
$\neg x$	逻辑运算 NOT ( $x$ 是一个布尔变量), 也表示比特运算: 按位求否 ( $x$ 是一个比特串)
$x \wedge y$	逻辑运算 AND ( $x, y$ 是布尔变量), 也表示比特运算: 按位求并 ( $x, y$ 是比特串)
$x \vee y$	逻辑运算 OR ( $x, y$ 是布尔变量), 也表示比特运算: 按位求或 ( $x, y$ 是比特串)
$x \oplus y$	逻辑运算 XOR ( $x, y$ 是布尔变量), 也表示比特运算: 按位求异或 ( $x, y$ 是比特串)
$(* \dots *)$	在算法或协议中非执行的注释部分
$\square$	证明、注释或例题的结束

## 第3章 概率论和信息论

### 3.1 引言

概率论和信息论是现代密码技术发展必不可少的工具。

概率论是进行安全性分析的基本工具。我们常常需要估计在确定的条件下,一个不安全事件发生的可能性有多大。比如,考虑第1章中的“电话掷币”协议,我们需要估计 Alice 能够以多大的概率找到已知单向函数  $f$  的一个碰撞(这个概率应该限定于一个很小的量),以及 Bob 以多大概率从已知的  $f(x)$  中求得  $x$  的奇偶性(这个概率应该非常接近  $1/2$ )。

信息论与概率论有着密切的联系。加密算法安全性的一个重要方面可以看成是“密文的不确定性”,即希望加密算法所输出的密文在整个密文消息空间中应该是随机分布的。香农用他所命名的熵的概念来量化信息的不确定性。历史上,人们希望实现密码中的高熵,以便挫败那些利用自然语言所包含的冗余度这一事实的密码分析技术,冗余度与自然语言中经常出现的一些已知图样有关。

最近,现代密码系统,特别是公钥密码系统,对概率行为的要求已经达到相当苛刻的程度:语义安全性。可以将其描述为下面的性质:给定一个语义安全的加密算法,如果 Alice 以相同的概率加密 0 或 1,然后将密文  $c$  送给 Bob,并让他回答所加密的内容,那么,如果 Bob 没有正确的解密密钥,他就不应该有一种算法策略,能够以比随机猜想更大的“优势”辨别这两种情况。我们注意到许多“教科书式”加密算法并不具备这种令人满意的性质。

#### 3.1.1 本章纲要

3.2 节~3.6 节所介绍概率论的基本概念,对于本书的使用来说是足够的。3.7 节~3.8 节介绍信息论。

### 3.2 概率论的基本概念

令  $S$  为一个任意确定的点的集合,称之为概率空间(或样本空间)。任意元素  $x \in S$  称为样点(也称为结果、简单事件或不可分事件;为了简单我们将只用点)。一个事件(也称为合成事件或可分事件)是  $S$  的一个子集,通常用一个大写字母表示(比如  $E$ )。一次实验或观察是一种从  $S$  中产生(取出)一个点的动作。一个事件  $E$  的发生就是一个试验产生某个点  $x \in S$ ,并满足  $x \in E$ 。

**例 3.1** 考虑一个实验,从一副公平扑克牌中抽取一张(其中“公平”是指随机地抽取一张牌)。这里是一些关于概率空间、点、事件和事件发生的实例。

1.  $S_1$ : 样本空间由 52 个点组成,其中每个点代表这副扑克牌中的一张。令事件  $E_1$  为“A”,(即,  $E_1 = \{A \spadesuit, A \heartsuit, A \diamondsuit, A \clubsuit\}$ )。

2.  $S_2 = \{\text{红}, \text{黑}\}$ , 令事件  $E_2 = \{\text{红}\}$ 。当所抽取的牌为红色时该事件发生。

3.  $S_3$ : 样本空间由 13 个点组成, 即 2, 3, 4,  $\dots$ , 10, J, Q, K, A。令事件  $E_3$  为“数字”。当所抽取的扑克为 2, 或 3, 或  $\dots$ , 或 10 时该事件发生。  $\square$

**定义 3.1 概率的经典定义** 假设一个实验可以从  $n = \#S$  个等可能的点中产生一个点, 并且每次实验必须产生一个点。令  $m$  表示事件  $E$  包含的点的数目, 那么称  $\frac{m}{n}$  为事件  $E$  发生的概率, 并记为

$$\text{Prob}[E] = \frac{m}{n}$$

**例 3.2** 在例 3.1 中:

$$1. \text{Prob}[E_1] = \frac{4}{52} = \frac{1}{13}.$$

$$2. \text{Prob}[E_2] = \frac{1}{2}.$$

$$3. \text{Prob}[E_3] = \frac{9}{13}. \quad \square$$

**定义 3.2 概率的统计定义** 假设在相同条件下进行了  $n$  次实验, 其中事件  $E$  发生了  $\mu$  次。如果对所有足够大的  $n$ ,  $\frac{\mu}{n}$  保持不变, 那么就说事件  $E$  的概率为  $\frac{\mu}{n}$ , 记为

$$\text{Prob}[E] \approx \frac{\mu}{n}$$

在 3.5.3 节中, 我们将看到定义 3.2 可以作为一个由其他几个直观概念推导出来的定理(大数定律的推论)。但是, 由于我们考虑到这个定义本身是很直观的, 故仍将其作为一个定义给出。

### 3.3 性质

1. 一个概率空间本身是一个事件, 称为**必然事件**。例如,  $S = \{\text{正面}, \text{反面}\}$ 。我们有

$$\text{Prob}[S] = 1$$

2. 用  $\emptyset$  表示不包括任何点的事件(即永不发生的事件)。比如,  $\text{黑} \diamond \in \emptyset$ 。这样的事件称为**不可能事件**。我们有

$$\text{Prob}[\emptyset] = 0$$

3. 任何事件  $E$  满足

$$0 \leq \text{Prob}[E] \leq 1$$

4. 如果  $E \subseteq F$ , 我们称事件  $E$  蕴含事件  $F$ , 并且

$$\text{Prob}[E] \leq \text{Prob}[F]$$

5. 用  $\bar{E} = S \setminus E$  表示  $E$  的补事件, 那么

$$\text{Prob}[E] + \text{Prob}[\bar{E}] = 1$$

### 3.4 基本运算

用  $E \cup F$  表示事件  $E$ 、 $F$  的和事件,表示这两个事件至少有一个发生;用  $E \cap F$  表示事件  $E$ 、 $F$  的积事件,表示这两个事件同时发生。

#### 3.4.1 加法规则

1.  $\text{Prob}[E \cup F] = \text{Prob}[E] + \text{Prob}[F] - \text{Prob}[E \cap F]$ 。
2. 如果  $E \cap F = \emptyset$ ,我们称这两个事件是互斥的或不相交的,并且

$$\text{Prob}[E \cup F] = \text{Prob}[E] + \text{Prob}[F]$$

3. 如果  $\bigcup_{i=1}^n E_i = S$  且  $E_i \cap E_j = \emptyset (i \neq j)$ , 则

$$\sum_{i=1}^n \text{Prob}[E_i] = 1$$

例 3.3 证明

$$\text{Prob}[E \cup F] = \text{Prob}[E] + \text{Prob}[F \cap \bar{E}] \quad (3.4.1)$$

因为  $E \cup F = E \cup (F \cap \bar{E})$ , 这里  $E$  和  $F \cap \bar{E}$  是互斥的, 由加法规则 2 可知 (3.4.1) 成立。□

**定义 3.3 条件概率** 设  $E$ 、 $F$  为两个事件, 且  $E$  的概率不为 0。在  $E$  发生的条件下  $F$  发生的概率称为已知  $E$  时  $F$  的条件概率, 记为

$$\text{Prob}[F|E] = \frac{\text{Prob}[E \cap F]}{\text{Prob}[E]}$$

**例 3.4** 考虑有两个孩子的家庭, 设字母  $g$  和  $b$  分别代表女孩和男孩, 并且第一个字母代表年龄较大的孩子。我们有四种可能  $gg$ 、 $gb$ 、 $bg$ 、 $bb$ , 这四个点构成空间  $S$ 。令每个点的概率均为  $\frac{1}{4}$ 。设事件  $E$  表示一个家庭有一个女孩, 事件  $F$  表示该家庭中两个孩子都是女孩。在已知事件  $E$  发生的条件下事件  $F$  发生的概率 (即  $\text{Prob}[F|E]$ ) 是多少?

事件  $E \cap F$  表示  $gg$  发生, 于是  $\text{Prob}[E \cap F] = \frac{1}{4}$ 。因为事件  $E$  表示  $gg$  或  $gb$  或  $bg$  发生, 因此  $\text{Prob}[E] = \frac{3}{4}$ 。所以由定义 3.3 可知,  $\text{Prob}[F|E] = \frac{1}{3}$ 。实际上, 我们可以期望, 在使事件  $E$  发生的所有家庭中, 有三分之一的家庭事件  $F$  会发生。□

**定义 3.4 独立事件** 事件  $E$ 、 $F$  是相互独立的, 当且仅当

$$\text{Prob}[F|E] = \text{Prob}[F]$$

#### 3.4.2 乘法规则

1.  $\text{Prob}[E \cap F] = \text{Prob}[F|E] \cdot \text{Prob}[E] = \text{Prob}[E|F] \cdot \text{Prob}[F]$ 。
2. 如果事件  $E$ 、 $F$  是相互独立的, 则

$$\text{Prob}[E \cap F] = \text{Prob}[E] \cdot \text{Prob}[F]。$$

**例 3.5** 考虑例 3.1, 假设事件  $E_1$  和  $E_2$  是相互独立的, 它们的概率分别为  $\frac{1}{13}$  和  $\frac{1}{2}$  (例 3.2)。由

于这两个事件是相互独立的,由“乘法规则 2”可知,这两个事件同时发生(抽取一张红色的  $A$ )的概率是  $\frac{1}{26}$ 。  $\square$

### 3.4.3 全概率定律

全概率定律是一个很有用的定理。

**定理 3.1** 如果  $\bigcup_{i=1}^n E_i = S$  且  $E_i \cap E_j = \emptyset (i \neq j)$ , 那么对任意事件  $A$  有

$$\text{Prob}[A] = \sum_{i=1}^n \text{Prob}[A|E_i] \cdot \text{Prob}[E_i]$$

**证明** 因为

$$A = A \cap S = \bigcup_{i=1}^n (A \cap E_i)$$

其中事件  $A \cap E_i$  和  $A \cap E_j (i \neq j)$  是互斥的,所以等号右边和事件的概率可以应用加法规则 2 求和来计算,而其中的每一项可以由“乘法规则 1”得到。  $\square$

全概率定律非常有用,我们会经常用它来计算在已知一些互斥事件(比如,典型互斥事件为  $E$  和  $\bar{E}$ )发生的条件下事件  $A$  发生的概率(或估算概率的界)。该公式之所以很有用,是因为通常计算条件概率  $\text{Prob}[A|E_i]$  要比直接计算  $\text{Prob}[A]$  更容易。

**例 3.6** (这个例子用到了数论的一些基本知识,对于理解本例题有困难的读者可以在学完第 6 章后再回过头来看这个例子。)

设  $p = 2q + 1$ , 其中  $p$  和  $q$  都是素数。考虑从集合  $S = \{1, 2, \dots, p-1\}$  中随机选取两个整数  $g$  和  $h$  (可重复)。令事件  $A$  表示“ $h$  由  $g$  生成”,即对某个  $x < p$  有  $h \equiv g^x \pmod{p}$  成立(等价地,这意味着“ $\log_g h \pmod{p-1}$  是存在的”)。对随机的  $g$  和  $h$  来说,事件  $A$  发生的概率是多少?

直接计算  $\text{Prob}[A]$  不是很方便。然而,如果先计算几个条件概率,然后再应用全概率定理,  $\text{Prob}[A]$  的计算就变得简单了。

用  $\text{ord}_p(g)$  表示  $g \pmod{p}$  的(乘法)阶,即满足  $g^i \equiv 1 \pmod{p}$  的最小的自然数  $i$ 。概率  $\text{Prob}[A]$  的值取决于下面四个互斥的事件。

- i)  $E_1: \text{ord}_p(g) = p-1 = 2q$ 。我们知道  $\text{Prob}[E_1] = \frac{\phi(2q)}{p-1} = \frac{q-1}{p-1}$  (这里  $\phi$  是 Euler 函数); 在  $S$  中恰好有  $\phi(2q) = q-1$  个阶为  $2q$  的元素。在这种情况下,任何  $h < p$  都可以由  $g$  产生( $g$  是集合  $S$  的生成元),因此我们有  $\text{Prob}[A|E_1] = 1$ 。
- ii)  $E_2: \text{ord}_p(g) = q$ 。与 i) 的情况相似,我们知道  $\text{Prob}[E_2] = \frac{q-1}{p-1}$ 。在这种情况下,当且仅当  $\text{ord}_p(h) | q$  时,  $h$  可以由  $g$  生成。因为在  $S$  中恰好有  $q$  个元素,其阶整除  $q$ ,因此我们有  $\text{Prob}[A|E_2] = \frac{q}{p-1} = \frac{1}{2}$ 。
- iii)  $E_3: \text{ord}_p(g) = 2$ 。因为只有一个元素的阶为 2,即  $p-1$ ,因此  $\text{Prob}[E_3] = \frac{1}{p-1}$ 。又因为只有 1 和  $p-1$  可以由  $p-1$  产生,所以我们有  $\text{Prob}[A|E_3] = \frac{2}{p-1}$ 。

iv)  $E_4: \text{ord}_p(g) = 1$ 。因为仅有元素 1 的阶为 1, 因此  $\text{Prob}[E_4] = \frac{1}{p-1}$ 。又因为只有 1 可以

由 1 生成。所以我们有  $\text{Prob}[A|E_4] = \frac{1}{p-1}$ 。

上面四个事件不仅是互斥的, 而且它们也包括了  $g$  的阶的所有可能情况。因此, 应用全概率定理可以求得  $\text{Prob}[A]$ :

$$\text{Prob}[A] = \frac{q-1}{p-1} + \frac{q-1}{2(p-1)} + \frac{2}{(p-1)^2} + \frac{1}{(p-1)^2} \quad \square$$

### 3.5 随机变量及其概率分布

在密码学中, 我们主要考虑定义在离散空间上的函数(例如作为密钥空间的整数区间, 或者有限代数结构, 如有限群或域)。设离散空间  $S$  包含有限个或者是可数个孤立的点  $x_1, x_2, \dots, x_n, \dots, x_{\#S}$ 。我们考虑一般的情况, 即  $S$  包含可数个点。这种情况下,  $\#S = \infty$ , 这将允许我们用一种渐进的方法进行算法和协议的计算复杂度分析(见 4.6 节)。

#### 定义 3.5 离散随机变量及其分布函数

- 1) 一个(离散)随机变量是一个实验的数字化结果。它是定义在一个(离散)样本空间上的函数。
- 2) 设  $S$  为一个(离散)概率空间,  $\xi$  为一个随机变量。 $\xi$  的(离散)分布函数是  $S \mapsto \mathbb{R}$  的一个函数, 以一个概率值

$$\text{Prob}[\xi = x_i] = p_i \quad (i = 1, 2, \dots, \#S)$$

列表为条件, 并满足下面的条件:

- i)  $p_i \geq 0$ ;
- ii)  $\sum_{i=1}^{\#S} p_i = 1$ 。

现在, 让我们看一下密码学中经常用到的两个离散概率分布。从现在起, 我们将省去“离散概率空间”、“离散概率分布”等词中的“离散”二字。我们讨论的所有情况都是离散的。

#### 3.5.1 均匀分布

密码学中最常用的随机变量服从均匀分布:

$$\text{Prob}[\xi = x_i] = \frac{1}{\#S} \quad (i = 1, 2, \dots, \#S)$$

**例 3.7** 设  $S$  表示最长为  $k$  比特(二进制数字)的非负数集合。依据均匀分布, 从  $S$  中随机取出一个数, 证明所取的数为  $k$  比特的概率是  $\frac{1}{2}$ 。

集合  $S = \{0, 1, 2, \dots, 2^k - 1\}$  可以分成两个不相交的子集  $S_1 = \{0, 1, 2, \dots, 2^{k-1} - 1\}$  和  $S_2 = \{2^{k-1}, 2^{k-1} + 1, \dots, 2^k - 1\}$ , 其中  $S_2$  包含了所有  $k$  比特的数, 且  $\#S_1 = \#S_2 = \frac{\#S}{2}$ 。应用“加法规则 2”, 我们有



$$\begin{aligned}
\text{Prob}[\text{抽样点} \in S_2] &= \text{Prob}\left[\bigcup_{i=2^{k-1}}^{2^k-1} \text{抽样点} = i\right] \\
&= \sum_{i=2^{k-1}}^{2^k-1} \text{Prob}[\text{抽样点} = i] \\
&= \sum_{i=2^{k-1}}^{2^k-1} \frac{1}{\#S} \\
&= \frac{\#S_2}{\#S} \\
&= \frac{1}{2}
\end{aligned}$$

□

尽管在密码学中也会经常用到,但该例子中关于“依据均匀分布,从(一个集合) $S$ 中随机地抽取(一个点) $p$ ”的表达还是相当长。所以,我们将这种长的表达缩短为“在 $S$ 中随机均匀地取出 $p$ ”,或者用更短的记法: $p \in_v S$ 。

### 3.5.2 二项式分布

假定一个实验只有两个结果,记为“成功”和“失败”(例如,抛一枚硬币只有两个结果,“正面”和“反面”)。独立地重复进行该实验,如果每一次实验结果仅有两种可能的点,且它们的概率在整个实验过程中保持不变,那么这样的实验就称为贝努利试验(bernoulli trials)。假设在任何一次试验中

$$\text{Prob}[\text{“成功”}] = p, \text{Prob}[\text{“失败”}] = 1 - p$$

那么

$$\text{Prob}[n \text{ 次试验有 } k \text{ 次结果为“成功”}] = \binom{n}{k} p^k (1-p)^{n-k} \quad (3.5.1)$$

其中 $\binom{n}{k}$ 表示“从 $n$ 个物中任取 $k$ 个”的不同取法数。

这里给出式(3.5.1)成立的原因。首先,事件“ $n$ 次试验结果为 $k$ 次‘成功’和 $n-k$ 次‘失败’”发生的不同方式数目等于“从 $n$ 个物体中任取 $k$ 个”的方法数,也就是说,该事件包含 $\binom{n}{k}$ 个点。其次,每一个点由 $k$ 次“成功”和 $n-k$ 次“失败”组成,我们可知该点的概率为 $p^k(1-p)^{n-k}$ 。

如果随机变量 $\xi_n$ 取值为 $0, 1, \dots, n$ ,且对每一个 $p, 0 < p < 1$ ,有

$$\text{Prob}[\xi_n = k] = \binom{n}{k} p^k (1-p)^{n-k} \quad (k = 0, 1, \dots, n)$$

那么,我们就称 $\xi_n$ 服从贝努利分布(binomial distribution)。与式(3.5.1)做一比较,可知贝努利试验服从贝努利分布。用 $b(k; n, p)$ 表示一个贝努利项(binomial term),其中 $k = 0, 1, \dots, n$ 且 $0 < p < 1$ 。

## 例 3.8

i) 将一枚公平的硬币抛掷 10 次。“正面出现”任意可能次数的概率是多少(即,出现 0 次,或 1 次, ..., 或 10 次)?

ii) “正面出现 5 次”的概率是多少?

iii) “正面出现少于或等于 5 次”的概率是多少?

对于(i), 因为该事件总是发生, 它的概率应该为 1。实际上, 应用“加法规则 2”, 我们有

$$\begin{aligned}\text{Prob}\left[\bigcup_{i=0}^{10} \text{正面出现 } i \text{ 次}\right] &= \sum_{i=0}^{10} \text{Prob}[\text{正面出现 } i \text{ 次}] \\ &= \sum_{i=0}^{10} \binom{10}{i} \left(\frac{1}{2}\right)^i \left(\frac{1}{2}\right)^{10-i} \\ &= (1+1)^{10} \left(\frac{1}{2}\right)^{10} \\ &= 1\end{aligned}$$

对于(ii), 我们有

$$\text{Prob}[\text{抛掷 10 次正面出现 5 次}] = \binom{10}{5} \left(\frac{1}{2}\right)^{10} = \frac{252}{1024} \approx 0.246$$

对于(iii), 必须将“正面出现”少于或等于 5 次的所有可能情况的概率加起来:

$$\text{Prob}\left[\bigcup_{i=0}^5 \text{抛 10 次正面出现 } i \text{ 次}\right] = \left(\frac{1}{2}\right)^{10} \sum_{i=0}^5 \binom{10}{i} \approx 0.623 \quad \square$$

图 3.1 给出了当  $p=0.5$  且  $n=10$  时的二项式分布图, 这两个参数就是例 3.8 所使用的参数。

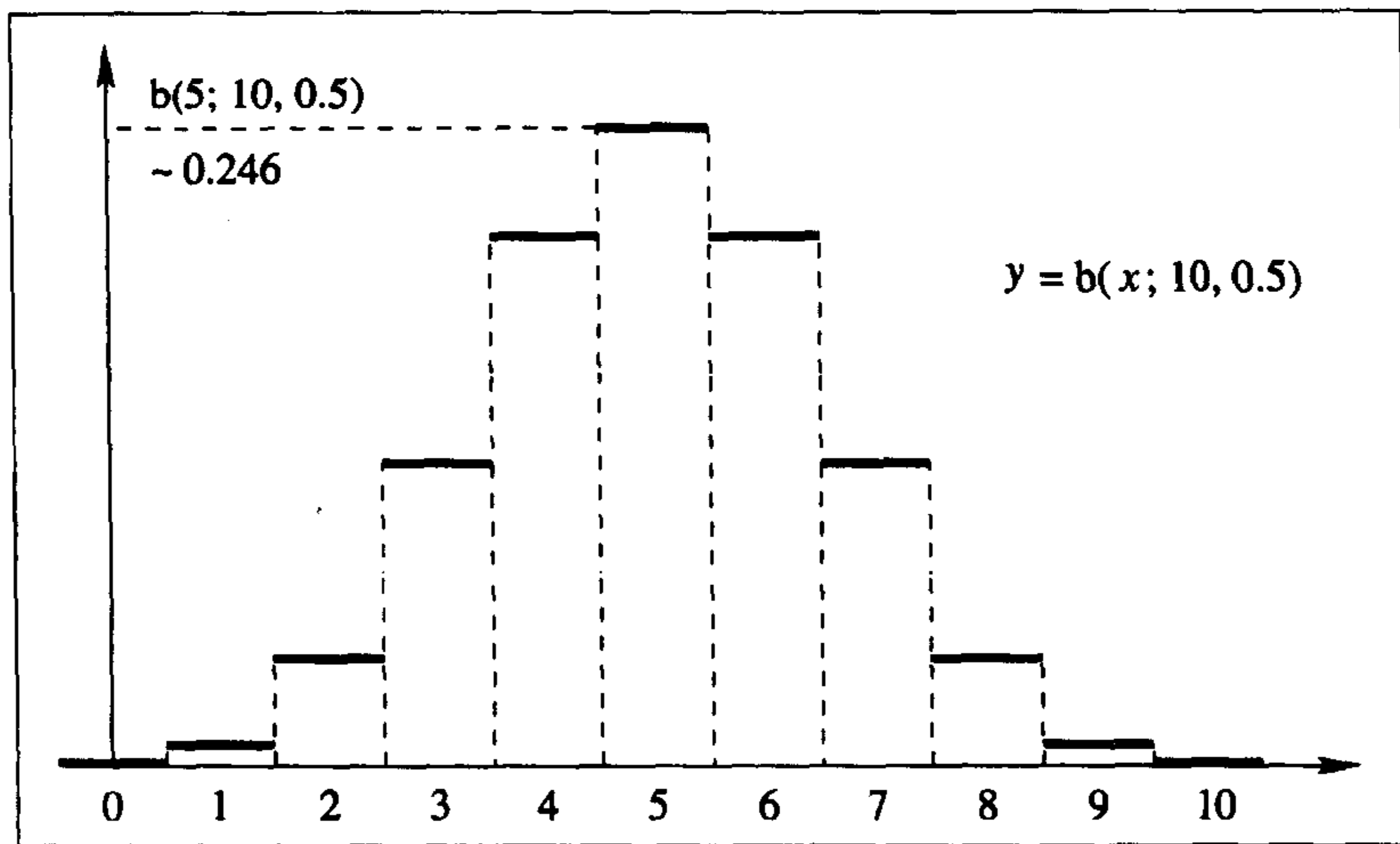


图 3.1 二项式分布

读者应当特别注意例 3.8(ii)和例 3.8(iii)之间的区别。前者是图 3.1 中的中心矩形区域, 而后者是左边六个矩形区域之和。

在二项式分布的应用中(例如, 在 4.4.1 节、4.4.5.1 节和 18.5.1 节中的应用), 恰好有  $r$  次“成功”的概率(如例 3.8(ii), 单独的一项)不如小于(大于)或等于  $r$  次“成功”的概率(如例 3.8(iii), 多个项之和)显得有用。而且, 有些项的和比起其他某些项之和来说要重要得多。现在, 我们就研究二项式分布中“重要的和”和“可忽略的和”。

## 3.5.2.1 中心项和尾部

相邻的二项式两项的比值是

$$\frac{b(k; n, p)}{b(k-1; n, p)} = \frac{(n-k+1)p}{k(1-p)} = 1 + \frac{(n+1)p - k}{k(1-p)} \quad (3.5.2)$$

对于等式右边的第二项来说,当  $k < (n+1)p$  时,其值为正;当  $k > (n+1)p$  时,其值为负。于是,当  $k < (n+1)p$  时,式(3.5.2)中的比值大于1;当  $k > (n+1)p$  时,该比值小于1。所以,在  $k$  增大到  $(n+1)p$  之前,  $b(k; n, p)$  随  $k$  的增大而增大;而当  $k > (n+1)p$  之后,  $b(k; n, p)$  随  $k$  的增大而减小。因此,在  $k = \lfloor (n+1)p \rfloor$  时,二项式项  $b(k; n, p)$  达到最大值。此二项式项

$$b(\lfloor (n+1)p \rfloor; n, p) \quad (3.5.3)$$

称为**中心项**(central term)。因为中心项达到了最大值,因此相应的点是具有“可能性最大的成功次数”的点。注意,当  $\lfloor (n+1)p \rfloor$  为整数时,式(3.5.2)中之比值为1,因此,在这种情况下有两个中心项,即  $b((n+1)p-1; n, p)$  和  $b((n+1)p; n, p)$ 。

设  $r > (n+1)p$ ,也就是说,  $r$  是位于具有“可能性最大的成功次数”的点右边的某个点。我们知道对所有的  $k \geq r$ ,  $b(k; n, p)$  随  $k$  的增大而减小。我们可以用  $r$  代替式(3.5.2)中最右边式子中的  $k$  来估算这种减小的速度,并得到

$$b(k; n, p) < b(k-1; n, p)s \quad \text{其中 } s = \frac{(n+1-r)p}{r(1-p)} < 1 \quad (3.5.4)$$

特别地,我们有

$$b(k; n, p) < b(r; n, p)s$$

注意,式(3.5.4)对所有的  $k = r+1, r+2, \dots, n$  成立,因此我们有

$$b(r+i; n, p) < b(r; n, p)s^i \quad i = 1, 2, \dots \quad (3.5.5)$$

对于  $r > np$ ,现在让我们来看多于或等于  $r$  次“成功”的概率上界,这个概率是

$$\text{Prob}[\xi_n \geq r] = \sum_{k=r}^n b(k; n, p) = \sum_{i=0}^{n-r} b(r+i; n, p) \quad (3.5.6)$$

由式(3.5.5),我们有

$$\text{Prob}[\xi_n \geq r] < b(r; n, p) \sum_{i=0}^{n-r} s^i < b(r; n, p) \sum_{i=0}^{\infty} s^i = b(r; n, p) \frac{1}{1-s}$$

将  $s$  替换回  $\frac{(n+1-r)p}{r(1-p)}$ ,我们有

$$\text{Prob}[\xi_n \geq r] < b(r; n, p) \frac{r(1-p)}{r - (n+1)p}$$

现在,我们注意到,中心项和  $b(r; n, p)$  之间仅有  $r - (n+1)p$  项,其中每一项均大于  $b(r; n, p)$ ,并且所有这些项之和仍小于1。因此,可以得到  $b(r; n, p) < (r - (n+1)p)^{-1}$ 。从而,我们最终得到

$$\text{Prob}[\xi_n \geq r] \leq \frac{r(1-p)}{(r - (n+1)p)^2}, \quad r > (n+1)p \quad (3.5.7)$$

式(3.5.7)中的界称为二项式分布函数的**右尾部**(right tail)。可以看出,如果  $r$  略微偏移中心点

$(n+1)p$ , 则分式(3.5.7)中的分母是非零的, 因此整个“右尾部”限制在一个大小为  $(np)^{-1}$  的量级内。所以, 右尾部是一个小量, 当  $n$  增大时减小并趋向于 0。

用类似的方法, 我们可以推导出左尾部(left tail)的界限:

$$\text{Prob}[\xi_n \leq r] \leq \frac{(n+1-r)p}{((n+1)p-r)^2}, \quad r < (n+1)p \quad (3.5.8)$$

其推导过程作为一个练习留给读者来完成(见习题 3.7)。

式(3.5.7)和式(3.5.8)的两个尾部看起来好像都限制在一个  $\frac{1}{n}$  量级的量内。但是, 我们应当注意, 式(3.5.7)和式(3.5.8)中的估算结果只是两个上界。尾部趋向于 0 的真实速度要比  $\frac{1}{n}$  趋向于 0 的速度快得多。下面的例子用具体数据说明了这个事实(也可以参看 18.5.1.1 节中的协议 18.4 的正确性和完整性)。

**例 3.9** 设  $p=0.5$ 。对于  $n$  的不同取值, 我们计算二项式分布函数关于点  $r=n(p-0.01)$  时左尾部的界限。

i) 对于  $n=1000$ , 相应的左尾部为

$$\text{Prob}[\xi < 490] \approx 0.253\ 33$$

ii) 对于  $n=10\ 000$ , 相应的左尾部变为

$$\text{Prob}[\xi < 4\ 900] \approx 0.022\ 21$$

iii) 对于  $n=100\ 000$ , 相应的左尾部成为一个无足轻重的量

$$\text{Prob}[\xi < 49\ 000] \approx 1.242\ 41 \times 10^{-10}$$

比较以上结果, 可以明显地看到, 尾部趋向于 0 的速度要比  $\frac{1}{n}$  趋向于 0 的速度快得多。

因为  $p=0.5$ , 所以它的分布密度函数是对称的(见图 3.1)。对一个对称分布来说, 如果左尾部和右尾部含有项的数目相同, 则左、右尾部的值相等。因此, 对于情形(iii), 两个尾部的 98 000 项(即总项数的 98%)之和几乎等于 0, 而可能性最大的成功次数的项(即在中心项附近, 占总项数 2%的 2001 项)之和几乎等于 1。□

### 3.5.3 大数定律

回忆定义 3.2: 在  $n$  次相同的试验中, 当  $n$  足够大时, 如果事件  $E$  总是发生  $\mu$  次, 那么  $\frac{\mu}{n}$  就是事件  $E$  发生的概率。

考虑在贝努利试验中, 如果“成功”的概率为  $p$ , 随即变量  $\xi_n$  是  $n$  次试验中“成功”的次数。那么  $\frac{\xi_n}{n}$  是  $n$  次试验中“成功”的平均次数。由定义 3.2,  $\frac{\xi_n}{n}$  应接近于  $p$ 。

现在, 我们来考虑对任意  $\alpha > 0$  (即  $\alpha$  为任意小但固定的正数),  $\frac{\xi_n}{n}$  大于  $p + \alpha$  的概率。显然, 这个概率为

$$\text{Prob}[\xi_n > n(p + \alpha)] = \sum_{i=n(p+\alpha)+1}^n b(i; n, p)$$

由式(3.5.7)可知

$$\text{Prob}[\xi_n > n(p + \alpha)] < \frac{1}{n\alpha} \quad (3.5.9)$$

于是

$$\text{Prob}[\xi_n > n(p + \alpha)] \rightarrow 0 \quad (n \rightarrow \infty) \quad (3.5.10)$$

类似地,我们可以得到

$$\text{Prob}[\xi_n < n(p - \alpha)] \rightarrow 0 \quad (n \rightarrow \infty)$$

因此,我们有(大数定律)

$$\lim_{n \rightarrow \infty} \text{Prob} \left[ \left| \frac{\xi_n}{n} - p \right| < \alpha \right] = 1$$

大数定律的这种表达形式也称为**贝努利定理**(Bernoulli's theorem)。现在,可以很明显地看到,定义3.2可以作为大数定律的一个推论。但是,由于考虑到它本身是很直观的,我们已将其以定义的形式给出。

### 3.6 生日悖论

对任意函数  $f: X \mapsto Y$ , 其中  $Y$  为包含  $n$  个元素的集合, 我们来解决下面的问题:

对于一个概率界限  $\epsilon$  (即  $0 < \epsilon < 1$ ), 找一个整数  $k$ , 使得对于  $k$  个两两互异的值  $x_1, x_2, \dots, x_k \in {}_U X$ ,  $k$  个函数值  $f(x_1), f(x_2), \dots, f(x_k)$  对某些  $i \neq j$  有

$$\text{Prob}[f(x_i) = f(x_j)] \geq \epsilon$$

即在  $k$  个函数值中, 以不小于  $\epsilon$  的概率发生碰撞。

要解决这个问题, 就需要找到一个整数  $k$ , 对任意的函数, 满足给定的概率下界。我们只需考虑具有随机性的函数: 将  $X$  中均匀分布的输入映射到  $Y$  中均匀分布的输出。显然, 只有具有这样随机性的函数才可以通过增大  $k$  值来满足给定的概率下界, 从而可在相同的概率界限下, 满足其他函数。因此,  $\#X > \#Y$  是必要的, 否则对某些函数来说, 可能根本不发生碰撞。

因此, 在上面的问题中, 我们可以假设函数值为  $n$  个互异的、等概的点。这样的函数值可以用一个模型表示: 从装有  $n$  个不同颜色小球的袋子中取一个球, 记下该球的颜色, 然后再将其放回袋子。那么, 上面的问题就是找到一个整数  $k$ , 至少出现一次颜色匹配的概率为  $\epsilon$ 。

取第一个球时颜色不受限制。并令  $y_i$  表示第  $i$  次取出的小球的颜色, 第二次取出的小球不和第一次取出的小球颜色相同, 于是  $y_2 \neq y_1$  的概率为  $1 - 1/n$ ;  $y_3 \neq y_1$  且  $y_3 \neq y_2$  的概率为  $1 - 2/n$ ; 等等, 以此类推。取第  $k$  个小球还未发生碰撞的概率为

$$\left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right)$$

当  $n$  足够大且  $x$  相对较小时, 我们知道

$$\left(1 + \frac{x}{n}\right)^n \approx e^x$$

或者

$$1 + \frac{x}{n} \approx e^{x/n}$$

因此,

$$\left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right) = \prod_{i=1}^{k-1} \left(1 + \frac{-i}{n}\right) \approx \prod_{i=1}^{k-1} e^{-i/n} = e^{-\frac{k(k-1)}{2n}}$$

最右边的等式是对指数求高斯和得到的。

这就是选取  $k$  个球而不发生碰撞的概率。因此至少发生一次碰撞的概率为

$$1 - e^{-\frac{k(k-1)}{2n}}.$$

令这个值等于  $\epsilon$ , 我们有

$$e^{-\frac{k(k-1)}{2n}} \approx 1 - \epsilon$$

或者

$$k^2 - k \approx 2n \log \frac{1}{1 - \epsilon}$$

即

$$k \approx \sqrt{2n \log \frac{1}{1 - \epsilon}} \quad (3.6.1)$$

因此, 对一个映射到  $Y$  上的随机函数, 在已知概率  $\epsilon$  的条件下, 为了发生碰撞仅需要计算这么多的函数值。从式(3.6.1)可以看到, 即使  $\epsilon$  的值非常重要(即非常接近于 1),  $\log \frac{1}{1 - \epsilon}$  的值仍然比较小, 因此通常  $k$  与  $\sqrt{n}$  成比例。

如果考虑  $\epsilon = 1/2$ , 那么

$$k \approx 1.177 4 \sqrt{n} \quad (3.6.2)$$

在式(3.6.1)和式(3.6.2)式中给出的  $k$  与  $n$  的平方根关系说明, 对一个输出空间大小为  $n$  的随机函数, 我们只需计算大约  $\sqrt{n}$  个函数值, 就能以一个不可忽略的概率发现一个碰撞。

这一事实对密码系统和密码协议的设计具有深远的影响。例如, 将一组数据(如一个密钥或一条消息)作为某个密码函数(典型的随机函数)的原像隐藏, 如果该数据的平方根不够大, 那么就可以通过随机计算函数值来找出这组数据。这种攻击称为平方根攻击或生日攻击。第二种叫法源于下面与之类似的“悖论现象”: 在式(3.6.2)中, 取  $n = 365$ , 所以  $k \approx 22.49$ ; 即为了以大于 50% 的概率从一个房间中找到有两个人的生日相同, 在该房间中只需有 23 人即可。这一结果初看上去似乎不大直观。

### 3.6.1 生日悖论的应用: 指数计算的 Pollard 袋鼠算法

设  $p$  为素数。在一定的条件下(将在第 5 章讨论), 模指数函数  $f(x) = g^x \pmod{p}$  本质上是一个随机函数。即对于  $x = 1, 2, \dots, p-1$ , 函数值  $f(x)$  在整数区间  $[1, p-1]$  范围内任意变化。这个函数具有单向性: 计算  $y = f(x)$  容易(运用算法 4.3), 而对几乎所有  $y \in [1, p-1]$ , 求函数的逆, 即求  $x = f^{-1}(y)$ , 非常困难。由于这个原因, 该函数在密码学中有广泛应用。

在某些情况下, 对于  $y = f(x)$ , 我们知道存在某些  $a$  和  $b$ , 有  $x \in [a, b]$ 。显然可以通过计算  $f(a), f(a+1), \dots$  在穷尽  $b-a$  步之前找到  $x$ 。如果  $b-a$  太大, 那么这种穷搜索方法不实用。但是, 如果  $\sqrt{b-a}$  是一个易处理的值(例如,  $b-a \approx 2^{100}$ , 因此  $\sqrt{b-a} \approx 2^{50}$ , 这是一个易处



理的量级),那么生日悖论在 $\sqrt{b-a}$ 步中求 $f(x)$ 的逆时起作用。Pollard发现了这种方法[240];他称这种算法为指数计算的 $\lambda$ 算法或袋鼠方法。过一会儿就会清楚这种命名的意义所在。

Pollard用两个袋鼠为例描述他的算法。一只驯养的袋鼠 $T$ ,另一只是野生的袋鼠 $W$ 。已知 $f(x) = g^x \pmod{p}$ 求未知指数 $x$ 的问题可以模型化为用 $T$ 追捕 $W$ 。这一点是通过让这两只袋鼠沿着下面的方式跳跃完成的。令 $S$ 为包含 $J(J = \lfloor \log_2(b-a) \rfloor)$ ,因此很小)个元素的整数集合:

$$S = \{s(0), s(1), s(2), \dots, s(J-1)\} = \{2^0, 2^1, 2^2, \dots, 2^{J-1}\}$$

袋鼠每一次跳跃的距离为 $S$ 中随机的一个数,每只袋鼠都随身携带一个里程表来计算它跳过的总距离。

$T$ 从一个已知点 $t_0 = g^b \pmod{p}$ 开始跳。因为 $T$ 是驯服的袋鼠,这个已知点 $b$ 可以认为是本垒(家宅)的所在。它的路线为

$$t(i+1) = t(i) \cdot g^{s(t(i) \pmod{J})} \pmod{p}, \quad i = 0, 1, 2, \dots \quad (3.6.3)$$

让 $T$ 跳 $n$ 步后停止。我们将在后面确定 $n$ 应为多大。在跳了第 $n$ 次后, $T$ 携带的里程表记录着它目前跳过的距离

$$d(n) = \sum_{i=0}^n s(t(i) \pmod{J})$$

运用 $T$ 的里程表所记录的距离,将式(3.6.3)重新表达为

$$t(n) = g^{b+d(n)} \pmod{p}$$

$W$ 从一个隐藏在 $w_0 = g^x \pmod{p}$ 中的未知点开始跳。该未知点是 $x$ ,这也是 $W$ 是野生的原因。它的路线为

$$w(j+1) = w(j) \cdot g^{s(w(j) \pmod{J})} \pmod{p}, \quad j = 0, 1, 2, \dots \quad (3.6.4)$$

$W$ 携带的里程表也记录着它目前跳过的距离

$$D(j) = \sum_{k=0}^j s(w_k \pmod{J})$$

类似于对 $T$ 足迹的表达,运用 $W$ 的里程表所记录的距离将式(3.6.4)重新表达为

$$w(i) = g^{x+D(i)} \pmod{p}$$

显然,两只袋鼠的足迹 $t(i)$ 和 $w(j)$ 是两个随机函数。前者的范围为 $i$ 个点的集合,后者的范围为 $j$ 个点的集合。根据生日悖论,在 $T$ 和 $W$ 分别大约跳

$$n \approx \sqrt{b-a}$$

步内,对某个 $\xi \leq n$ 和 $\mu \leq n$ ,碰撞 $t(\xi) = w(\mu)$ 应当发生。也就是 $T$ 和 $W$ 跳在了同一个点,这可以想像为 $W$ 跳进了 $T$ 所设置的陷阱中,现在就抓获了 $W$ 。如果两个袋鼠随机跳跃的次数超过 $\sqrt{b-a}$ ,那么碰撞发生的概率就很快趋向于1。

当碰撞 $t(\xi) = w(\mu)$ 发生时,观察式(3.6.3)和式(3.6.4),可以得到 $t(\xi+1) = w(\mu+1)$ , $t(\xi+2) = w(\mu+2), \dots$ ,等等。也就是说,对某些整数 $m \approx n$ , $w(m) = t(n)$ 最终会出现。可以想像,碰撞等式 $t(\xi) = w(\mu)$ 代表希腊字母 $\lambda$ 的两条腿相遇的点,在该点之后,两只袋鼠以

相同的路线跳跃直至最终发现  $w(m) = t(n)$  (回忆  $T$  固定地跳  $n$  步)。这就解释了以  $\lambda$  命名该算法的原因。

当发现该碰撞后,有

$$g^x = g^{b+d(n-1)-D(m-1)} \pmod{p}$$

即已经得到

$$x = b + d(n-1) - D(m-1)$$

因为有两个里程表  $d(m-1)$  和  $D(n-1)$ , 可以用它们所累加的“里”数来计算  $x$ 。当两只袋鼠跳到同一点后, 可能又跳了很长的距离, 因此所求得的指数可能为  $x + o$ , 其中  $o$  满足  $g^o \pmod{p} = 1$ 。如果是这种情况的话, 将  $x + o$  当做要计算的指数值并不影响要求的结果。

这是一个概率算法, 这就意味着该算法可能因找不到碰撞而失败(也就是说, 不能计算出目标指数值)。然而, 依据 3.6 节中所讨论的重要的碰撞概率可知, 算法失败的概率可以控制在一个足够小的量内。通过一个已知的偏移量  $\delta$  重新设置  $W$  的起始位置来重复该算法, 这样重复几次后, 该算法就可结束。

$\lambda$  算法能够实用的条件是  $\sqrt{b-a}$  不是很大。因此, 设置  $n = \sqrt{b-a}$  ( $T$  跳跃的次数), 算法所需的时间与计算  $\sqrt{b-a}$  次模指数运算成比例。算法的空间需求很小: 只需要存储  $J = \lfloor \log(b-a) \rfloor$  个元素。时间限制  $\sqrt{b-a}$  意味着利用该算法求大指数是不实用的。Pollard 解释这个局限性的原因是因为袋鼠不能跳越大洋。

### 3.7 信息论

香农关于消息源的熵(entropy)的定义[264, 265]用来衡量这个源所含信息量的多少。这个量度以源输出的所有可能的消息集上的概率分布函数形式给出。

设  $L = \{a_1, a_2, \dots, a_n\}$  为由  $n$  个不同符号组成的语言。假设信源  $S$  以独立的概率即  $\text{Prob}[a_1], \text{Prob}[a_2], \dots, \text{Prob}[a_n]$  分别输出这些符号, 并且这些概率满足

$$\sum_{i=1}^n \text{Prob}[a_i] = 1 \quad (3.7.1)$$

信源  $S$  的熵为

$$H(S) = \sum_{i=1}^n \text{Prob}[a_i] \log_2 \left( \frac{1}{\text{Prob}[a_i]} \right) \quad (3.7.2)$$

式(3.7.2)中定义的熵函数  $H(S)$  所取的值, 我们称之为“每个信源输出的比特数”。

我们通过安排一个简单的工作来解释熵函数: 所考虑的信源  $S$  是无记忆的, 必须记录信源  $S$  的输出。一种直接方法就是记录信源  $S$  的输出。但是, 由式(3.7.1)知道,  $S$  的每次输出均为已知的  $n$  个符号  $a_1, a_2, \dots, a_n$  中的一个, 记录这些已知的东西是枯燥的, 而且效率不高。因此, 问题是, 如何才能有效地记录  $S$  输出中我们所关心的东西?

设  $S$  以  $k$  连续串的形式输出这些符号, 即  $S$  输出的是包含  $k$  个符号的单词

$$a_{i_1} a_{i_2} \dots a_{i_k}, \quad 1 \leq i_k \leq n$$

令  $L_k$  表示记录  $S$  输出的、包含  $k$  个符号的单词所需最少的比特数。我们有下面的定理用于衡量  $L_k$  的值。

**定理 3.2 Shannon** [264, 265]

$$\lim_{k \rightarrow \infty} \frac{L_k}{k} = H(S)$$

**证明** 对所有的整数  $k > 0$ , 下面的“三明治”型关系式成立:

$$kH(S) \leq L_k \leq kH(S) + 1$$

定理所述的是其极限形式。 □

换句话说, 为记录信源  $S$  的每个输出, 所需的最小平均比特数为  $H(S)$ 。

### 3.7.1 熵的性质

如果  $S$  以概率 1 输出某个符号, 例如  $a_1$ , 则熵函数  $H(S)$  有最小值 0。这是因为

$$H(S) = \text{Prob}[a_1] \log_2 \left( \frac{1}{\text{Prob}[a_1]} \right) = \log_2 1 = 0$$

这种情况说明, 当我们确信信源  $S$  确定地仅输出  $a_1$ , 那我们何必还要浪费一些比特来记录它呢?

如果  $S$  以相等的概率  $1/n$  输出每个符号, 即  $S$  是一个均匀分布的随机信源, 则熵函数  $H(S)$  达到最大值  $\log_2 n$ 。这是因为在这种情况下,

$$H(S) = \frac{1}{n} \sum_{i=1}^n \log_2 n = \log_2 n$$

这种情况也说明了下面一个事实: 因为  $S$  可以以相等的概率输出这  $n$  个符号中的任何一个符号, 我们至少要用  $\log_2 n$  比特来记录这  $n$  个数字中的任何一种可能。

最后, 可以认为  $H(S)$  是信源  $S$  每次输出所包含的不确定性或信息量。

**例 3.10** 考虑协议 1.1 (“电话掷币”)。不管是通过电话还是通过连接的计算机, 对 Alice 和 Bob 来说, 该协议都是协商一个随机比特。在该协议中, Alice 随机选取一个大的整数  $x \in {}_U\mathbb{N}$ , 通过单向函数  $f$  计算  $f(x)$ , 并将其送给 Bob, 最后, 在 Bob 随机猜测后披露  $x$ 。在 Bob 看来,  $x$  作为整数不应该被当做一条新的信息, 因为即使在接收到  $f(x)$  前他已经知道  $x$  是  $\mathbb{N}$  中的一个元素。Bob 仅利用 Alice 输出中的有用部分: 运用  $x$  的奇偶性来计算与 Alice 的输出相符的随机比特。因此, 我们有

$$\begin{aligned} H(\text{Alice}) &= \text{Prob}[x \text{ 为奇数}] \log_2 \left( \frac{1}{\text{Prob}[x \text{ 为奇数}]} \right) + \\ &\quad \text{Prob}[x \text{ 为偶数}] \log_2 \left( \frac{1}{\text{Prob}[x \text{ 为偶数}]} \right) \\ &= \frac{1}{2} \log_2 2 + \frac{1}{2} \log_2 2 = 1 \end{aligned}$$

也就是说, 尽管 Alice 的输出是一个大整数, 但她是一个每次输出 1 比特的信源。 □

如果 Alice 和 Bob 重复执行  $n$  次协议 1.1, 他们就能够协商一个  $n$  比特的串: 若 Bob 猜对一次, 则输出 1; 猜错一次输出 0。该协议的这种用法, 使得 Alice 和 Bob 都是一个每执行一次

协议输出 1 比特的信源。双方都相信所获得的比特串是随机的,因为任何一方都有她/他自己的随机输入,并且知道另一方无法控制其输出。

### 3.8 自然语言的冗余度

考虑一个输出为自然语言  $L$  中单词的信源  $S(L)$ 。假设  $L$  中平均每个单词包含  $k$  个字符。由香农定理(定理 3.2)可知,  $H(S(L))$  为信源  $S(L)$  一次输出的最小平均比特数(请记住  $S(L)$  的每次输出为  $k$  个字符的单词),所以语言  $L$  中每字符的最小平均比特数应该为

$$r(L) = \frac{H(S(L))}{k}$$

其中  $r(L)$  称为语言  $L$  的速率。设  $L$  为英语,香农计算的  $r(\text{英语})$  的范围为 1.0 到 1.5 比特/字母[267]。

设  $\Sigma = \{a, b, \dots, z\}$ , 那么有  $r(\Sigma) = \log_2 26 \approx 4.7$  比特/字母。 $r(\Sigma)$  称为具有字母表  $\Sigma$  的语言的绝对速率。通过比较  $r(\text{英语})$  和  $r(\Sigma)$ , 我们可以看到,英语的真实速率比其绝对速率小得多。

具有字母集  $\Sigma$  的语言  $L$  的冗余度为

$$r(\Sigma) - r(L) \text{ (比特/字符)}$$

因此,保守地令  $r(\text{英语}) = 1.5$ , 英语的冗余度为  $4.7 - 1.5 = 3.2$  比特/字母。用百分比表示,冗余率为  $3.2/4.7 \approx 68\%$ 。换句话说,英语单词中大约 68% 的字符都是多余的。这就意味着,在不丢失信息的前提下,可以将一篇英语文章压缩为原长的 32%。

自然语言的冗余度源于语言本身的一些已知经常出现的图样。例如,在英语中,字母  $q$  后面几乎总是跟着字母  $u$ ; “the”、“ing”和“ed”是其他几个已知图样的例子。自然语言的冗余度为密码分析(cryptanalysis)提供了一个重要工具,密码分析的目的是从密文中获取明文消息或密钥。

**例 3.11** 我们在第 1 章曾经提到,在本书中我们将研究密码算法和协议的多种攻击方法。在后面的章节里(第 14 章),我们将介绍和讨论对加密算法的四种攻击方法,这四种攻击方法都有很长的名字。它们是

- 对明文不可区分的被动攻击
- 选择明文模式下对明文不可区分的主动攻击
- 非适应性选择密文模式下对明文不可区分的主动攻击
- 适应性选择密文模式下对明文不可区分的主动攻击

这些攻击方式的完整意义将在第 14 章给予解释。这里仅需要指出关于这些攻击的以下两个事实:

1. 使用长的命名是很合适的,这是因为每一种这样的长名称攻击后面都传达了大量的信息。
2. 在第 14 章,我们将只讨论这四种攻击。

因为在第 14 章才要讨论这四种攻击,实际上,这些名称的熵可以低到 2 比特/名称那么低。但是,因为数字 0、1、2 和 3,以及其他一些单个的字符(如字母“a”,下标“i”、“j”,安全

参数“ $k$ ”,等等)都将在第14章中出现,为了惟一地区分这些攻击,实际上必须用多于两个比特的信息来命名这些攻击。

注意,不是在第14章中的任何部分都不使用串  $a_0, a_1, a_2, a_3$ ; 实际上可以将这四种攻击的长名称分别缩短为这四个字符串而不会引起任何混淆。因此,在第14章的范围内,这四种攻击名称的熵可以很合理地低到  $4.7 + 2 = 6.7$  (比特/名称)。其中 4.7 比特代表字母“ $a$ ”,而 2 比特用来表示数字 0, 1, 2, 3。

另一方面,通过简单地计算,读者可以求出这四个名称的平均长度为 62.75 (字母)。因此,每字母的平均比特数为  $6.7/62.75 < 0.107$ 。根据这个结果,可以进一步计算出这些长名称的冗余度(在第14章范围内):

$$\frac{6.7 - 0.107}{6.7} > 98\% \quad \square$$

因此,这些长的攻击名称是有非常多的冗余。

但是,对可证明强安全密码系统的研究范围不仅仅局限于第14章。因此,实际上,例 3.11 中的那些非常短的名称  $a_0, a_1, a_2, a_3$  太短而不能用来命名这些攻击(使用这么短的名称可能会造成理解上的含糊,令人感到不悦)。事实上,例 3.11 中的后三种攻击的名称通常分别缩短为 IND-CPA、IND-CCA 和 IND-CCA2。在第14章我们也采用这些名称。

最后,我们指出仅将后面三个长的名称缩短的原因是因为在我们的研究范围内,经常要讨论这三种攻击。对于“(明文不可区分的)被动攻击”,因为不常讨论这种攻击而且也很容易防止,我们就会觉得用这样长的名称是很自然的。

### 3.9 本章小结

在这一章里,我们仅对概率论和信息论进行了很初步的研究,但这些内容对于本书的使用而言是足够的。

在概率论方面,理解和熟悉基本的概念、性质和基本运算规则非常重要。要强调的是,很好地理解这些很基本的知识并不是一件困难的事,并且对我们会有很大的帮助。我们已经看到,许多有用的定理和工具,如全概率定律、大数定律和生日悖论等,都可以由几个基本的、直观性质和规则推导出。

在本书的其余部分,将常常遇到关于条件概率、全概率定律、二项式分布和生日悖论的应用(我们已经看到 Pollard  $\lambda$  算法就是生日悖论的一个很好的应用)。在这些应用中,我们将会逐渐熟悉这些工具。

我们也学习了信息论简单的基本知识。现在我们知道,信源的熵就是衡量从信源发出的消息中所包含的信息量,或消息的随机(不可预测)程度。

### 习题

3.1 按顺序抛两个骰子。求以下事件发生的概率:

- i) 点数之和为 7, 1, 并且小于或等于 12;
- ii) 第二个的点数小于第一个的点数;

iii) 至少有一个的点数为 6;

iv) 在已知第一个的点数为 6 的条件下,第二个点数也为 6。

3.2 在前面的问题中,求已知第一个的点数为 3 的条件下,点数之和大于或等于 8 的概率。

3.3 已知 4.5% 的总人口和 0.6% 的女性是色盲,求占总人口 49.9% 的男性中色盲所占的百分比是多少?

提示:应用全概率定律。

3.4 假设  $\theta$  在  $[-\pi/2, \pi/2]$  内均匀分布,分别求  $\sin\theta \leq 1/2$  和  $|\sin\theta| \leq 1/2$  的概率。

3.5 在一个数集中,有四分之一的数为平方数。从该集合中随机取出 5 个数,求所取的数中多数为平方数的概率。

提示:类似于例 3.8 (iii),将平方数个数  $\geq 3$  的所有情况加起来。

3.6 什么是二项式分布函数的(左,右)尾部?

3.7 推导式(3.5.8),二项式分布函数“左尾部”的上界。

3.8 为什么说定义 3.2 可以看做是由大数定律所推导出的定理?

3.9 设  $n = pq$ ,其中  $p$  和  $q$  为位数大致相等的、不相同的大素数。我们知道,对任意的  $a < n$  和  $\gcd(a, n) = 1$ ,  $a^{p+q} = a^{n+1} \pmod{n}$ 。证明:可以在  $n^{1/4}$  步内分解  $n$ 。

提示:注意到  $p + q \approx n^{1/2}$ ,可以应用 Pollard 的  $\lambda$  算法求  $a^{p+q} \pmod{n}$  的指数  $p + q$ ,然后用  $p + q$  和  $pq$  分解  $n$ 。

3.10 在“电话掷币”协议中,Alice 均匀随机地选取一个大整数,在 Alice 端进行计算时信源的熵是多少? 如果让 Bob 来计算,熵又是多少?

3.11 例 3.11 中,我们已经相应于四个极短名称  $a_0, a_1, a_2, a_3$  计算了四种攻击的长名称的冗余度,这些攻击将在第 14 章进行介绍。现在,在第 14 章范围内,计算下面四种进行了合理缩短的攻击名称的冗余度:

- Passive IND-Attack
- IND-CPA
- IND-CCA
- IND-CCA2



## 第4章 计算复杂性

### 4.1 引言

如果一个随机变量服从均匀分布并且与任意给定信息独立,则不存在任何方法将该均匀随机变量通过任何“计算”方式与任何其他信息联系起来。这正是惟一的无条件安全(或信息论安全)加密方案,即一次一密的安全基础,一次一密也就是将一个均匀的随机串(称为密钥串)和一个消息串逐比特混合(见 7.3.3 节)。密钥串和消息串之间的独立性要求两个串一样长,遗憾的是,这成为一次一密加密方案实用的一个几乎不可逾越的障碍。

不过(也有些讽刺意味),我们还算“幸运”。在写作本书的时候,我们(也因此对密码破译者)可广泛利用的计算设备和计算方法所基于的计算概念还不算太强有力。如果有两条信息其中一条只是看起来是随机的,而事实上它与另一条是完全相关的(如许多密码体制中的明文和密文消息),要通过计算将它们联系起来,至今我们并不怎么成功。因此,现代密码学将它的安全性基础建立在所谓的复杂性理论模型之上。此类密码体制的安全性是以某些问题难处理的各种假设为条件的。这里,“难处理”表示广泛可利用的一般计算方法不能有效地解这些问题。

应该指出,我们的“幸运”也许只是暂时的。已经出现了一种新的、更强大的计算模型:量子信息处理(QIP, quantum information processing)。在这种新的计算模型下,指数量级的计算可以通过所谓量子的“叠加”(Super-position)态的操作并行地完成。这样一来,构成基于复杂性理论密码学安全基础的许多有用的难问题将无困难可言,也就是说,它们将变得毫无用处。例如,如果处理的整数大小差不多,那么利用量子计算机,整数的相乘和分解花的时间也差不多,因此,Rivest、Shamir 和 Adleman 的著名密码体制(RSA)[248](见 8.5 节)也将被淘汰。不过,在写作本书的时候,QIP 技术离实用还有很远的距离。目前分解合数的记录是 15(参阅[302]),这是最小的无平方因子的奇合数。

因此,目前我们还不必太担心 QIP 技术。本章其余部分将介绍“不够强”的传统计算模型和现代密码学的复杂性理论基础。

#### 4.1.1 本章概述

4.2 节介绍图灵计算模型。4.3 节介绍确定性多项式时间类、几个有用的确定性多项式时间算法和复杂性度量的表示。4.4 节和 4.5 节介绍非确定性多项式时间(NP)问题的两个子类。第一个子类(见 4.4 节)是概率多项式时间问题,它进一步分为四类可有效求解的问题(见 4.4.2 节和 4.4.5 节)。第二个子类(见 4.5 节)要有内部知识才能有效求解,它在基于复杂性理论的现代密码学中具有重要作用。4.6 节介绍没有任何多项式界的复杂性概念。4.7 节以判定性问题作为非多项式界问题的实例:多项式时间不可区分性。最后,4.8 节讨论计算复杂性理论和现代密码学的关系。

## 4.2 图灵机

为了精确定义有效程序(即算法)这一概念,图灵构思了一种称为**图灵机**(Turing machine)的计算设备,把它作为一个计算原型,但却是非常通用的计算模型。这里要介绍的计算复杂性材料沿用了图灵机计算模型。下面介绍图灵机的一个变型,这对我们学习计算复杂性的目的来说已经足够了。对图灵机的一般描述参阅[9]中的1.6节。

在我们的变型里,图灵机(见图4.1)由有限状态控制单元、 $k (\geq 1)$ 条纸带(tape)以及同样数量的读写头(tapehead)组成。有限控制单元控制磁头读写纸带的操作,每个读写头访问一条纸带(称为它的纸带),沿着纸带或左或右地移动完成这一操作。每一条纸带分成无限个单元(cell)。图灵机求解一个问题时,读写头扫描一个有有限个字符的串。该串从纸带最左边的单元开始按顺序存放在纸带上,每个字符占用一个单元,右边剩下的是空白(blank)单元,该串称为问题的一个输入。扫描从含有输入的纸带左端开始,同时图灵机赋一个初态(initial state)。任何时刻图灵机都只有一个读写

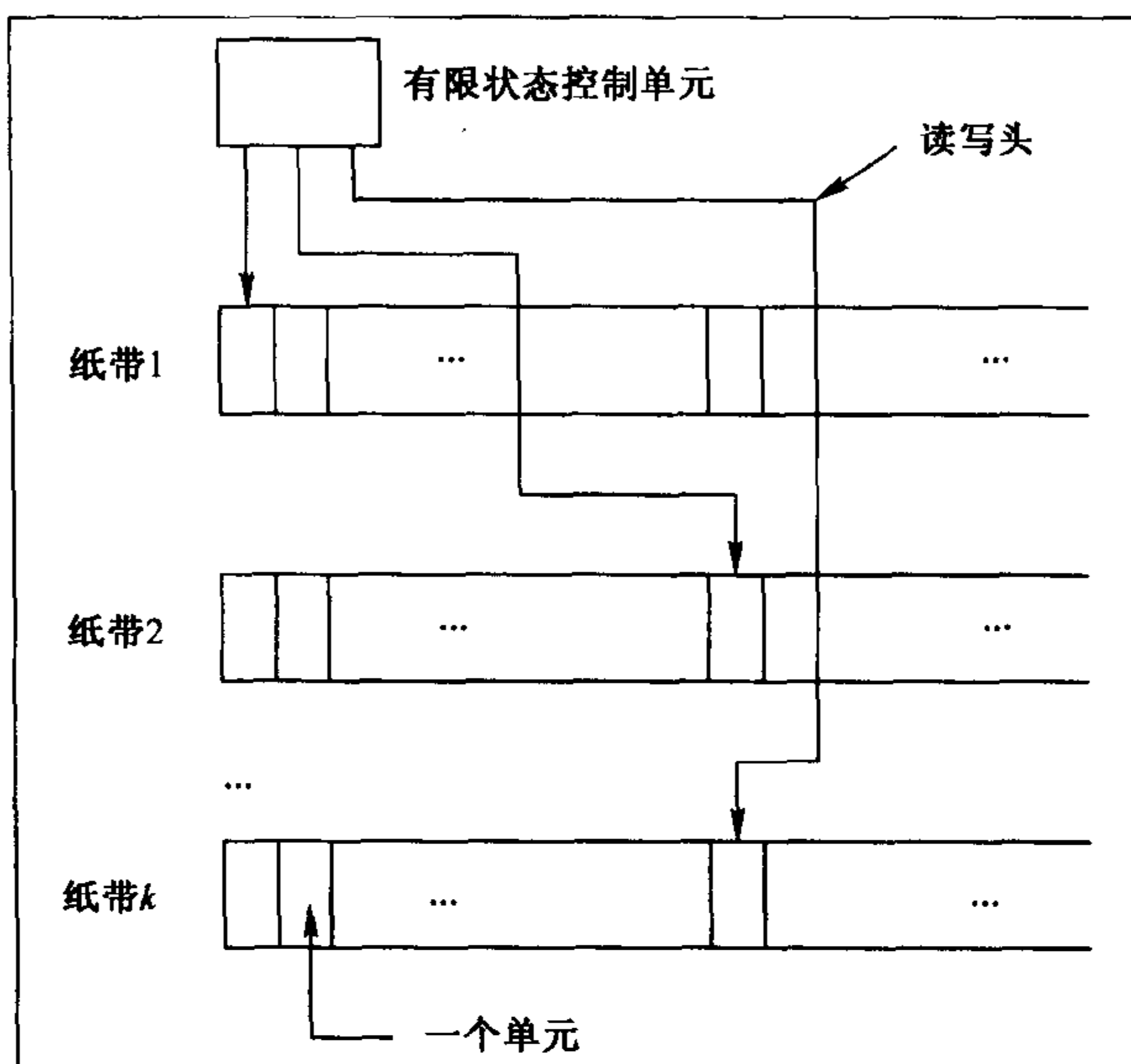


图 4.1 图灵机

头访问它的纸带。读写头对它的纸带的一次访问称为一个(合法)移动(move)。如果图灵机从初始状态开始,一步接一步地合法移动,完成对输入串的扫描,最终满足了终止条件而停下来,则称图灵机识别了该输入。否则,图灵机在某一点没有合法移动,它会没有识别输入就停下来。图灵机识别的一个输入称为一种可识别语言(language)的一个实例。

对给定的问题,图灵机可以由它的有限控制单元的功能完全描述。这样的功能可以用一张表的形式给出,列出图灵机每个状态的下一步。稍后将给出一个问题作为例子来描述图灵机(见下面的例4.1)。

为识别一个输入,图灵机  $M$  在停下来之前所移动的步数称为  $M$  的运行时间或  $M$  的时间复杂性,记为  $T_M$ 。很明显,  $T_M$  可以表示为函数  $T_M(n): \mathbb{N} \rightarrow \mathbb{N}$ , 其中  $n$  是输入实例的长度或规模,也就是说,当  $M$  在初始状态时,它就是组成输入串的字符数。显然,  $T_M(n) \geq n$ 。除了对时间的要求外,  $M$  还有对空间的要求  $S_M$ , 它是  $M$  在写操作中读写头访问的纸带单元数。  $S_M$  可以表示为函数  $S_M(n): \mathbb{N} \rightarrow \mathbb{N}$ , 称为  $M$  的空间复杂性。

下一节将看到一个具体的图灵机。

## 4.3 确定性多项式时间

我们从可以由确定性图灵机在**多项式时间**内识别的语言类开始。函数  $p(n)$  是整数上关

于  $n$  的一个多项式,如果它具有下列形式:

$$p(n) = c_k n^k + c_{k-1} n^{k-1} + \cdots + c_1 n + c_0 \quad (4.3.1)$$

其中  $k$  和  $c_i (i=0,1,2,\cdots,k)$  是常整数且  $c_k \neq 0$ 。当  $k > 0$  时,前者称为多项式  $p(n)$  的次数,记为  $\deg(p(n))$ ;后者称为多项式  $p(n)$  的系数。

**定义 4.1 P类** 我们记  $P$  表示具有下列特征的语言类。语言  $L$  在  $P$  中,如果存在一个图灵机  $M$  和一个多项式  $p(n)$  使得对任意非负整数  $n$ ,  $M$  可以在时间  $T_M(n) \leq p(n)$  内识别任意实例  $I \in L$ , 其中  $n$  是表示实例  $I$  规模的整数参数。我们称  $L$  是可以在多项式时间内识别的。

粗略地说,可以在多项式时间内识别的语言总是很“容易的”。换句话说,多项式时间图灵机被认为总是很“有效的”(我们将在 4.4.6 节定义“容易”或“有效”这一概念)。这里解释一下“总是”的含义。识别  $P$  中语言的图灵机都是确定的(deterministic)。确定性图灵机的输出结果完全取决于输入和初始状态。也就是说,对同样的输入和初始状态运行一个确定性图灵机两次,两次的输出结果是相同的。

应该注意到,在定义 4.1 中,全称限制条件“任意实例  $I \in L$ ”和“对任意非负整数  $n$ ”非常重要。在计算复杂性研究中,认为一个问题已经解决了,仅当该问题的任何实例都可以用同一个图灵机求解(即同一种方法)。惟有如此,该方法才是充分通用的,从而事实上可以认为是一种方法,我们用下面的例子来说明。

**例 4.1 DIV3 语言** 设 DIV3 是被 3 整除的非负整数集,证明  $\text{DIV3} \in P$ 。

我们通过构造一个在多项式时间内识别 DIV3 的单带图灵机来完成证明。

首先注意到,如果将输入写成一个三进制表示的整数,也就是说,输入是  $\{0,1,2\}$  中字符的一个串,那么识别该问题就变得非常简单:输入  $x$  属于 DIV3 当且仅当  $x$  的最后一位是 0。因此,构造的图灵机只需向右不断移动直到一个空白字符,然后停下来,当且仅当最后一个非空字符是 0,回答“是”。显然,这个图灵机可以在移动实例规模的步数内识别该实例,因此  $\text{DIV3} \in P$ 。

不过,我们希望证明  $\text{DIV3} \in P$  与输入表示无关。这只需证明输入表示为二进制的情形。设这样的图灵机称为 Div3。Div3 的有限状态控制按照图 4.2 一步步移动。

当前状态	纸带上的符号	下一步移动	下一个状态
$q_0$ (初态)	0	右	$q_0$
	1	右	$q_1$
	空白	“响铃”或终止	
$q_1$	0	右	$q_2$
	1	右	$q_0$
$q_2$	0	右	$q_1$
	1	右	$q_2$

图 4.2 图灵机 Div3 的运行

现在论述由图 4.2 中功能所定义的图灵机 Div3 足以识别 DIV3 中的所有实例。

首先,注意到要识别一个二进制串  $x \in \text{DIV3}$  是否成立,Div3 只需三个状态,分别对应于它(它的读写头)完成扫描串  $3k$ 、 $3k+1$  和  $3k+2 (k \geq 0)$  的情形。最小输入实例 0 约定 Div3 在

完成扫描输入串 0 时必定处于初态(不失一般性,设初态为  $q_0$ )。不失一般性,在完成扫描输入串 1 时,我们可以指定 Div3 为状态  $q_1$ ,完成扫描输入串 2 时,Div3 为状态  $q_2(=(10)_2)$ <sup>①</sup>。对任何二进制表示的非负整数  $a$ ,后面跟一个 0(或 1)得到值  $2a$ (或  $2a+1$ )。因此,完成对  $a=3k$ (当 Div3 初态为  $q_0$  时)的扫描后,由于在该点完成扫描  $2a=6k=3k'$ ,当下一步扫描到字符 0 时,Div3 必定仍在状态  $q_0$ ;下一步扫描到字符 1 时必定移动到  $q_1$ ,因为在该点完成扫描  $2a+1=6k+1=3k'+1$ 。类似地,完成对  $a=3k+1$ (当 Div3 在状态  $q_1$  时)的扫描后,当完成扫描  $2a=6k+2=3k'+2$  时,Div3 必定移动到  $q_2$ ;当完成扫描  $2a+1=6k+3=3k'$  时,必定移动到  $q_0$ 。 $a=3k+2$  还有两种情形: $2a=6k+4=3k'+1$ (Div3 从  $q_2$  移动到  $q_1$ )和  $2a+1=6k+5=3k'+2$ (Div3 停留在  $q_2$ )。因此,对任意  $k \geq 0$ ,三个状态分别对应 Div3 完成扫描  $3k$ 、 $3k+1$  和  $3k+2$ 。现在,一旦读写头遇到特殊字符“空”,只有在状态  $q_0$  的 Div3 才设置为响铃并停止移动(表示终止并回答“是”),从而识别出输入  $3k$ ;对其他两种状态,Div3 没有合法移动,因此没有识别就终止了。最后,显然  $T_{\text{Div3}}(n)=n$ 。因此,Div3 确实可以在多项式时间内识别 DIV3 语言。□

#### 例 4.2

- i) 比特串  $10101(=(21)_{10})$  是可识别的;Div3 识别该串需要  $T_{\text{Div3}}(|10101|)=|10101|=5$  次移动。
- ii) 比特串  $11100001(=(225)_{10})$  也是一个可识别的实例;Div3 识别该串需要  $T_{\text{Div3}}(|11100001|)=|11100001|=8$  次移动。
- iii) 比特串  $10(=(2)_{10})$  是不可识别的;Div3 确定它不可识别需要 2 次移动。□

### 4.3.1 多项式时间计算性问题

根据定义, $\mathcal{P}$  是多项式时间语言识别问题类。语言识别问题是一个判定性问题。对任意可能输入,一个判定性问题要求输出为“是”或“否”。但是  $\mathcal{P}$  类是非常普遍的,包括多项式时间的计算性问题。对任意可能的输入,计算性问题要求输出比“是/否”更一般的答案。既然图灵机可以向纸带写字符,那么它当然能够输出比“是/否”更一般的答案。

举一个例子,我们可以另外设计一种图灵机,它不仅能够识别任意实例  $x \in \text{DIV3}$ ,而且在识别  $x$  后还能输出  $x/3$ 。将这个新的图灵机命名为 Div3-Comp。实现 Div3-Comp 的一个非常简单的方法是将输入写成三进制表示。那么,输入是 DIV3 中的一个实例,当且仅当它最后一位是 0,在识别该输入之后,图灵机的输出就是输入纸带上的内容去掉最后一个 0,除非纸带上全是 0。如果坚持要求 Div3-Comp 只输入输出二进制数,那么 Div3-Comp 可以按下列方式实现。首先将输入  $x$  从二进制转化成三进制表示,一旦获得了三进制表示的  $x/3$ ,再转化为二进制表示作为输出。很明显,转换可以机械地逐位进行,需要  $C \cdot |x|$  次移动,其中,  $C$  是一个常数。现在我们知道

$$T_{\text{Div3-Comp}}(|x|) \leq C \cdot |x|$$

其中  $C$  是一个常数。从这个例子可以清楚地看到, $\mathcal{P}$  类必定包括 Div3-Comp 可以解决的问题。

① 我们用  $(a_1 a_2 \dots a_n)_b$  表示一个数的  $b$  进制表示,其中  $a_i < b$  且  $i=1,2,\dots,n$ ;如果不引起混淆, $b=10$  和  $b=2$  时省略  $b$ 。

$\mathcal{P}$ 包括多项式时间的计算性问题的一般证明可以如下给出。在所谓的冯·诺伊曼模型(即我们熟悉的现代计算机模型[229])中,一个计算设备包含一个计数器、一个存储器和一个中心处理单元(CPU),可以执行下面的基本指令,称为微指令:

- Load: 将存储位中的值加载到(CPU 中的)一个寄存器中
- Store: 将寄存器中的值存入存储器一个位置
- Add: 将两个寄存器的值相加
- Comp: 将一个寄存器中的值取补(通过“Add”实现减法)
- Jump: 给计数器赋一个新值
- JumpZ: 当一个寄存器的值为0时执行“Jump”(实现条件分支)
- Stop: 终止

众所周知(例如参阅[9]的1.4节),在冯·诺伊曼计算机上,上述很小的微指令集足以在冯·诺伊曼计算机上构造求解任何算术问题的算法(不过,注意“任意算术问题”不表示不考虑实例的规模,稍后将进一步讨论这个问题)。可以证明(例如[9]中的定理1.3),上述指令集中的每一条微指令都可以用图灵机在多项式时间内仿真。因此,冯·诺伊曼计算机可以在多项式时间内解决的问题(它表示算法用到的微指令数是关于算法输入规模的一个多项式)一定可以用图灵机在多项式时间内解决。这是因为,对任意两个多项式  $p(n)$  和  $q(n)$ ,  $p(n)$ 、 $q(n)$ 、 $p(q(n))$  和  $q(p(n))$  的任意组合结果还是关于  $n$  的多项式。注意在(简化的)微指令集里我们故意没包括乘法和除法,两个规模  $n$  的数相乘可以通过  $n$  次加来实现,因此其总的开销可以表示为  $n \times \text{cost}(\text{Add})$ 。除和乘的开销一样,因为它是重复相当于加一个相反数的减法。

需要提一下,基于图灵机的计算模型和基于冯·诺伊曼计算机的计算模型之间有一个不很重要的区别。根据定义4.1,我们认为一个问题在图灵机上是易处理的,当且仅当在同一台机器上它的任何实例都是易处理的(“一台图灵机可以处理其中的所有问题!”)。图灵机上解一个问题的开销由问题的规模来度量,以问题的整个规模平均的方式进行,没有必要事先确定问题规模的界。例4.1中的Div3机明显地表明了这一点。由于度量的这一特点,我们说基于图灵机的计算模型使用的是平均开销度量来衡量复杂性。相反,寄存器和逻辑电路是冯·诺伊曼计算机的基本组成模块,它的规模是固定的。因此,冯·诺伊曼计算机上易处理的问题必须预先确定规模;同理,实例规模越大,用来解决它需要的机器就越大。通常,不同规模的机器解同一个问题的平均开销度量并不一样。因此我们说基于电路的计算模型(冯·诺伊曼计算机就基于这样的模型)使用的是非均匀开销度量。但是,到目前为止,均匀开销度量和非均匀开销度量之间的差别并没有导致新的复杂性类,或导致已知的类失败。这就是为什么说这个差别不重要。

本章其余部分通常忽略判定性问题和计算性问题的差别以及图灵机、现代计算机、程序或算法之间的差别。判定性或计算性问题统称为问题,而机器、计算机、程序或算法统称为方法或算法。有时候,我们会回过头来描述语言识别问题,只有在那时候才会再用图灵机作为我们的基本工具。

### 4.3.2 算法与计算复杂度表示

现在我们来学习三个非常有用的多项式时间算法。通过学习这些算法,我们将(i)熟悉本

书中用来写算法和协议的一种编程语言,(ii)统一用来表示算法和协议计算复杂性的一些记号和约定,(iii)确定密码学中经常要用到的一些算术运算的时间复杂性。

图灵机提供了一种通用的计算模型,以及度量程序计算复杂性的一个准确概念,在上面已经对此做了解释。但是,一般并不希望按照这种原型机器来描述算法,甚至也不希望按照现代计算机的微指令(即在4.3.1节中描述的指令集)来描述。为了清楚有效地描述算法和数学命题,我们将使用一种高级编程语言,称为“准编程语言”,它和一些通用的高级编程语言诸如Pascal或C很接近,由于它具有不言自明的清晰特征,理解起来不会有任何困难。

#### 4.3.2.1 最大公因子

我们要研究的第一个算法是著名的欧几里得算法,计算最大公因子(算法4.1)。 $\gcd(a, b)$ 表示整数 $a$ 和 $b$ 的最大公因子,定义为整除 $a$ 和 $b$ 的最大整数。

---

##### 算法 4.1 求最大公因子的欧几里得算法

输入 整数  $a > b \geq 0$ ;

输出  $\gcd(a, b)$ 。

1. if  $b = 0$  return( $a$ );
  2. return( $\gcd(b, a \bmod b)$ ).
- 

在算法4.1中,“ $a \bmod b$ ”表示 $a$ 除以 $b$ 的余数(4.3.2.5节将正式定义模运算,并给出模运算一些有用的性质)。条件 $a > b \geq 0$ 只是为了便于表示。实现时,这个条件可以根据 $a, b$ 的绝对值调整它们的位置得到满足,当 $|a| < |b|$ 时调用 $\gcd(|b|, |a|)$ 。

现在来考察算法4.1是如何工作的。对整数 $a > b$ ,总可以写成

$$a = bq + r \quad (4.3.2)$$

其中整数 $q \neq 0$ ( $a$ 除以 $b$ 的商)且 $0 \leq r < b$ ( $a$ 除以 $b$ 的余数)。根据定义, $\gcd(a, b)$ 整除 $a$ 和 $b$ ,式(4.3.2)表明它必然也整除 $r$ ,所以 $\gcd(a, b)$ 等于 $\gcd(b, r)$ 。由于( $a$ 除以 $b$ 的)余数 $r$ 表示成了 $a \bmod b$ ,我们得到

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

这是在算法4.1中用到的事实,也就是说, $\gcd(a, b)$ 由 $\gcd(b, a \bmod b)$ 递归确定。递归调用 $\gcd$ 计算出下列等式,每一个都具有式(4.3.2)的形式并由两个输入值之间的除得到:

$$\begin{aligned} a &= bq_1 + r_1 \\ b &= r_1q_2 + r_2 \\ r_1 &= r_2q_3 + r_3 \\ &\vdots \\ r_{k-3} &= r_{k-2}q_{k-1} + r_{k-1} \\ r_{k-2} &= r_{k-1}q_k + r_k \end{aligned} \quad (4.3.3)$$



其中  $r_k = 0$  (它产生第一步要用到的终止条件),  $q_1, q_2, \dots, q_k, r_1, r_2, \dots, r_{k-1}$  是非负整数。由于  $r_k = 0$ , 式(4.3.3)的最后一个等式意味着  $r_{k-1}$  整除  $r_{k-2}$ , 在倒数第二个等式中, 它必定整除  $r_{k-3}$ , 如此类推, 最终, 如式(4.3.3)中第一个等式所示,  $r_{k-1}$  必定整除  $a$  和  $b$ 。其他等式中的任何余数都不具有这个性质(这就是为什么称它们为余数而不是因子的原因; 只有  $r_{k-1}$  是式(4.3.3)的最后一个等式中的因子)。因此,  $r_{k-1}$  事实上是  $a$  和  $b$  的最大公因子, 即  $r_{k-1} = \gcd(a, b)$ 。

例如,  $\gcd(108, 42)$  包含下列递归调用:

$$\gcd(108, 42) = \gcd(42, 24) = \gcd(24, 18) = \gcd(18, 6) = \gcd(6, 0) = 6$$

#### 4.3.2.2 扩展欧几里得算法

算法 4.1 丢弃了所有的中间商数。如果在  $\gcd(a, b)$  的计算中把它们累积起来, 就可以得到比  $\gcd(a, b)$  更多的东西。我们来看能得到什么。

式(4.3.3)中的第一个等式可以写成

$$a + b(-q_1) = r_1$$

等式两边同时乘以  $q_2$  得到

$$aq_2 + b(-q_1q_2) = r_1q_2$$

由这个等式和式(4.3.3)的最后两个等式可以推出

$$a(-q_2) + b(1 + q_1q_2) = r_2 \quad (4.3.4)$$

其他的可以用同样的方法进行计算。一般地, 对  $i = 1, 2, \dots, k$ , 我们有

$$a\lambda_i + b\mu_i = r_i \quad (4.3.5)$$

其中  $\lambda_i, \mu_i$  是式(4.3.4)中的整数, 是那些中间商数的某种累计结果。在 4.3.2 节中我们已经知道, 采用这种方法计算最终会得到  $r_k = 0$ , 那么有

$$a\lambda_{k-1} + b\mu_{k-1} = r_{k-1} = \gcd(a, b) \quad (4.3.6)$$

一个输入为  $a, b$ 、输出为  $\lambda_{k-1}, \mu_{k-1}$  的算法满足式(4.3.6), 则称之为扩展欧几里得算法。扩展欧几里得算法在本书后面对计算模整数除法有着广泛的用途。现在来描述这一算法, 也就是说, 寻找一种通用的方法来累计那些中间商数。

观察式(4.3.3)中的等式, 记  $r_{-1} = a, r_0 = b, \lambda_{-1} = 1, \mu_{-1} = 0, \lambda_0 = 0, \mu_0 = 1$ 。那么对  $i = 1, 2, \dots, k-1$ , 由式(4.3.3)的第  $i$  个等式,  $r_{i-1}, r_i$  和  $r_{i+1}$  具有如下关系:

$$r_{i+1} = r_{i-1} - r_i q_{i+1} \quad (4.3.7)$$

用式(4.3.5)替换式(4.3.7)右边的  $r_{i-1}$  和  $r_i$ , 我们导出

$$r_{i+1} = a(\lambda_{i-1} - q_{i+1}\lambda_i) + b(\mu_{i-1} - q_{i+1}\mu_i) \quad (4.3.8)$$

比较式(4.3.8)和式(4.3.5), 得到(对  $i = 0, 1, \dots, k-1$ )

$$\begin{aligned} \lambda_{i+1} &= \lambda_{i-1} - q_{i+1}\lambda_i \\ \mu_{i+1} &= \mu_{i-1} - q_{i+1}\mu_i \end{aligned} \quad (4.3.9)$$

这两个等式给出了计算最大公因子时累计那些中间商数的一般方法(见算法 4.2)。

**算法 4.2 扩展欧几里得算法**

输入 整数  $a, b$ : 满足  $a > b \geq 0$ ;

输出 整数  $\lambda, \mu$ , 满足  $a\lambda + b\mu = \gcd(a, b)$ 。

1.  $i \leftarrow 0; r_{-1} \leftarrow a; r_0 \leftarrow b;$   
 $\lambda_{-1} \leftarrow 1; \mu_{-1} \leftarrow 0; \lambda_0 \leftarrow 0; \mu_0 \leftarrow 1;$  (\* 初始化 \*)
2. while( $r_i = a\lambda_i + b\mu_i \neq 0$ ) do (\*  $r_i = a\lambda_i + b\mu_i$  恒成立 \*)
  - (a)  $q \leftarrow r_{i-1} \div r_i;$  (\*  $\div$  表示整数除法 \*)
  - (b)  $\lambda_{i+1} \leftarrow \lambda_{i-1} - q\lambda_i; \mu_{i+1} \leftarrow \mu_{i-1} - q\mu_i;$  (\* 求商数的和 \*)
  - (c)  $i \leftarrow i + 1;$
3. return( $(\lambda_{i-1}, \mu_{i-1})$ ).

**注释 4.1** 为了以一种易于理解的方式揭示算法 4.1 和算法 4.2 的工作原理,我们选择了牺牲效率。在下面两节(4.3.2.3 节~4.3.2.4 节),我们将分析它们的时间复杂度,并将结果和计算最大公因子最广为人知的时间复杂度进行比较。□

**4.3.2.3 欧几里得算法的时间复杂度**

现在来计算两个欧几里得算法的时间复杂度。显然,算法 4.1 的递归调用次数等于算法 4.2 的循环次数,也就是式(4.3.3)中的  $k$ 。

考虑  $a > b$  的情形,对  $i = 1, 2, \dots, k-1$ , 观察式(4.3.7), 得到下面两种情形之一:

$$|r_i| < |r_{i-1}| \quad (4.3.10)$$

或

$$|r_{i+1}| < |r_{i-1}| \quad (4.3.11)$$

进一步注意到  $r_{i+1} < r_i$ , 所以式(4.3.10)蕴含式(4.3.11), 也就是说, 式(4.3.11)恒成立。这表明  $k$  的最大值以  $2 \cdot |a|$  为界。如果将模运算视为基本运算, 花费一个时间单位, 那么算法 4.1 实现  $\gcd$  的时间复杂度以  $2 \cdot |a|$  为界, 这是关于  $a$  的规模的线性函数。

**定理 4.1** 计算最大公因子  $\gcd(a, b)$  至多要执行  $2\max(|a|, |b|)$  次模运算。因此, 算法 4.1 和算法 4.2 将在  $2\max(|a|, |b|)$  次循环内终止。□

G. Lamé(1795 ~ 1870)首先证明了定理 4.1 的第一个论断, 被认为是关于计算复杂性理论证明的第一个定理([178]的第 35 页)。

式(4.3.3)的一系列等式是由一系列除法得到的, 表明了最大公因子计算内在的顺序特性。自从欧几里得发现这一算法(即算法 4.1)以来, 还没有发现任何重大改进可以减少这一似乎必要的顺序步骤。

**4.3.2.4 计算复杂度的两种表示**

当衡量一个算法的计算复杂度时, 通常很难并且也没有必要确切指出复杂性度量界表示式中的常系数。阶号(order notation)使得度量复杂性的任务可以更容易一些。

**定义 4.2 阶号** 用  $O(f(n))$  表示函数  $g(n)$ , 满足存在常数  $c > 0$  和自然数  $N$ , 对所有的  $n \geq N$ , 有  $|g(n)| \leq c|f(n)|$ 。

使用记号  $O()$  可以将算法 4.1 和算法 4.2 的复杂性表示为  $O(\log a)$ 。注意在这个表达式中, 我们用  $\log a$  代替了  $|a|$  而没有明确说明对数的底数(尽管习惯上认为省略的底数是自然底数  $e$ )。读者可以验证, 在阶号下, 任意底数  $b > 1$  都给出正确的复杂性度量表示(见习题 4.10)。

到目前为止, 我们一直认为计算一次模运算花费一个单位时间, 也就是说, 它的时间复杂度是  $O(1)$ 。然而事实上, 一般情况下, 模运算“ $a \bmod b$ ”包含除法  $a \div b$ , 为了保留商数, 这实际是用算法 4.2 完成的。因此, 模运算的时间复杂度和除法的时间复杂度一样, 取决于两个操作的规模。从实用角度来看(要了解“实用”的含义, 见 4.4.6 节的最后部分), 用  $O(1)$  表示一次除法的时间对敏感的资源管理来说太粗略了。

对阶号的一个简单修正是采用按位计算法来度量算术运算。在按位计算中, 所有的变量要么是值 0, 要么是 1, 运算是逻辑运算而不是算术运算: 也就是  $\wedge$  (与)、 $\vee$  (或)、 $\oplus$  (异或) 和  $\neg$  (非)。

**定义 4.3 按位计阶号 (Bitwise Order Notation)** 记  $O_B()$  表示在按位计算模式下的  $O()$ 。

在位模式下, 两个整数  $i$  和  $j$  之间的加和减需要  $\max(|i|, |j|)$  次按位运算, 即时间为  $O_B(\max(|i|, |j|))$ 。直观上, 两个整数  $i$  和  $j$  之间的乘和除需要  $|i| \cdot |j|$  次按位运算, 即要时间  $O_B(\log i \cdot \log j)$ 。应该指出, 对于乘法(和除法), 如果使用快速傅里叶变换(FFT), 可以得到更低的时间复杂度  $O_B(\log(i+j) \log \log(i+j))$ 。不过, 这个更低的复杂度只是渐近的, 与一个大得多的常系数(和 FFT 的开销相关)相关, 实际上对相对小的操作数(如现代密码学用的数)可能带来更高的复杂度。因此本书不考虑用 FFT 来实现乘除法, 这样, 我们就只用直观上的复杂性来衡量乘法和除法。

现在用更精确的按位计阶号  $O_B()$  来表示算法 4.1 和算法 4.2 的时间复杂度。在定理 4.1 中, 我们已经知道, 对  $a > b$ ,  $\gcd(a, b)$  可以在时间  $O(\log a)$  内计算出来, 给定两个输入值不超过  $a$ , 模运算或除法花费时间  $O_B((\log a)^2)$ , 那么算法 4.1 和算法 4.2 的时间复杂度都是  $O_B((\log a)^3)$ 。

现在我们应该回忆起注释 4.1: 我们选择了通过牺牲效率来给出更易于理解其工作原理的算法。事实上, 所牺牲的效率相当大!

精心实现这两个算法要用到下述两个事实:

- i) 模运算或除法使  $a = bq + r$  的时间代价为  $O_B((\log a)(\log q))$ 。
- ii) 式(4.3.3)中的商  $q_1, q_2, \dots, q_k$  满足

$$\sum_{i=1}^k \log q_i = \log \prod_{i=1}^k q_i \leq \log a \quad (4.3.12)$$

因此, 经过精心实现, 计算最大公因子总的时间可以不超过

$$\sum_{i=1}^k O_B((\log a)(\log q_i)) \leq O_B((\log a)^2)$$

算法 4.1 和算法 4.2 相似算法的精心实现在[80]的第 1 章中可以找到。

本书的其余部分我们将用最著名的结果  $O_B((\log a)^2)$  表示用欧几里得算法或扩展欧几里得算法计算最大公因子总的时间复杂度。

#### 4.3.2.5 模算术

我们将要学习的一个重要的多项式时间确定性算法是计算模指数的一个算法。模指数在公钥密码学中广泛使用。首先简短地讨论一下模算术(熟悉模算术的读者可以跳过本节)。

**定义 4.4 模运算** 给定整数  $x$  和  $n > 1$ , “ $x \pmod n$ ”是  $x$  除以  $n$  的余数, 即一个非负整数  $r \in [0, n-1]$ , 对某个整数  $k$  满足

$$x = kn + r$$

**定理 4.2 模运算性质** 设整数  $x, y, n \neq 0$  满足  $\gcd(y, n) = 1$ , 模运算具有下列性质:

1.  $(x + y) \pmod n = [(x \pmod n) + (y \pmod n)] \pmod n$ ;
2.  $(-x) \pmod n = (n - x) \pmod n = n - (x \pmod n)$ ;
3.  $(x \cdot y) \pmod n = [(x \pmod n) \cdot (y \pmod n)] \pmod n$ ;
4. 记  $y^{-1} \pmod n$  表示  $y$  模  $n$  的乘法逆。它是  $[1, n-1]$  中一个惟一确定的整数, 满足  $(y \cdot y^{-1}) \pmod n = 1$ 。

**证明** 我们只证明 1 和 4, 2 和 3 留做习题(见习题 4.4)。

对  $0 \leq r, s \leq n-1$ , 我们可以写成  $x = kn + r, y = \ell n + s$ 。

对于 1, 我们有

$$\begin{aligned} (x + y) \pmod n &= [(kn + r) + (\ell n + s)] \pmod n \\ &= [(k + \ell)n + (r + s)] \pmod n \\ &= (r + s) \pmod n \\ &= [(x \pmod n) + (y \pmod n)] \pmod n \end{aligned}$$

对于 4, 因为  $\gcd(y, n) = 1$ , 对输入  $y$  和  $n$  运用扩展欧几里得算法(算法 4.2), 我们得到整数  $\lambda$  和  $\mu$  满足

$$y\lambda + n\mu = 1 \quad (4.3.13)$$

不失一般性, 有  $\lambda < n$ , 因为否则可以用  $\lambda \pmod n$  代替  $\lambda$ , 对某个  $k$  用  $y\lambda + \mu$  代替  $\mu$ , 同时保证式(4.3.13)仍然成立。根据定义 4.4,  $y\lambda \pmod n = 1$ 。因此我们找到了  $y^{-1} = \lambda < n$  为  $y$  模  $n$  的乘法逆。下面证明  $y^{-1}$  在  $[1, n-1]$  中的惟一性。假定  $y \pmod n$  存在另外一个乘法逆, 记为  $\lambda' \in [1, n-1], \lambda' \neq \lambda$ 。我们有

$$y(\lambda - \lambda') \pmod n = 0$$

即存在某个整数  $a$  使得

$$y(\lambda - \lambda') = an \quad (4.3.14)$$

我们知道, 对某个整数  $a$  有  $y = \ell n + 1$ 。因此, 对某个整数  $b$ , 等式(4.3.14)是

$$(\ell n + 1)(\lambda - \lambda') = an$$

或者

$$\lambda - \lambda' = bn$$

这与假设  $\lambda, \lambda' \in [1, n-1], \lambda \neq \lambda'$  矛盾。  $\square$

与有理数  $\mathbb{Q}$  中的除法一样, 除以一个数的模  $n$  定义为乘以除数的逆, 当然, 和有理数  $\mathbb{Q}$  中的情形一样, 这要求逆存在。因此, 对满足  $\gcd(y, n) = 1$  的任意  $y$ , 把  $x/y \bmod n$  写成  $xy^{-1} \bmod n$ 。

由于计算  $y^{-1}$  涉及到运用扩展欧几里得算法, 所以它需要  $O_B((\log n)^2)$  时间。因此, 模  $n$  除法的时间复杂度为  $O_B((\log n)^2)$ 。

定理 4.2 表明, 模算术与整数算术非常相似。不难看出, 加法和乘法服从下列交换律和结合律(其中“ $\circ$ ”表示加或者乘):

$$a \circ b \bmod n = b \circ a \bmod n \quad (\text{交换律})$$

$$a \circ (b \circ c) \bmod n = (a \circ b) \circ c \bmod n \quad (\text{结合律})$$

最后要指出, 在模运算  $x \bmod n$  (见定义 4.4) 的定义中,  $k$  的值 ( $x$  除以  $y$  的商) 并不重要。因此在等式

$$x \bmod n = y \bmod n \quad (4.3.15)$$

中, 不需要关心  $x$  和  $y$  是否相差一个  $n$  的倍数。这样, 上述等式总是写成

$$x \equiv y \pmod{n}$$

或

$$x \pmod{n} \equiv y$$

称等式 (4.3.15) 的这种表示方式为模  $n$  同余 (congruence), 或者说  $x$  与  $y$  模  $n$  同余。

#### 4.3.2.6 模指数

对  $x, y < n$ , 模指数  $x^y \pmod{n}$  按照整数幂的通常定义,  $x$  自乘  $y$  次, 但要模  $n$ :

$$x^y \stackrel{\text{def}}{=} \underbrace{xx \cdots x}_y \pmod{n}$$

设  $y \div 2$  表示  $y$  除以 2 取整, 即

$$y \div 2 = \begin{cases} y/2 & \text{如果 } y \text{ 是偶数} \\ (y-1)/2 & \text{如果 } y \text{ 是奇数} \end{cases}$$

然后应用模乘法的“结合律”, 我们有

$$x^y = \begin{cases} (x^2)^{y \div 2} & \text{如果 } y \text{ 是偶数} \\ (x^2)^{y \div 2} x & \text{如果 } y \text{ 是奇数} \end{cases}$$

上述计算给出了著名的所谓“重复平方-乘”运算计算模指数的算法。算法重复下列步骤: 将指数除以 2, 执行一次平方, 如果指数是奇数, 再额外执行一次乘法。算法 4.3 描述了这一方法的递归实现。

应当注意到算法 4.3 由递归定义带来的一个性质: 执行“return”语句意味着“return”语句后面的步骤永远都执行不到。这是因为 `return(“value”)` 语句使程序以“value”返回到了 `mod_exp` 当前调用点。因此在算法 4.3 中, 如果执行了第二步就执行不到第三步。

**算法 4.3 模指数**

输入 整数  $x, y, n: x > 0, y \geq 0, n > 1$ ;

输出  $x^y \pmod n$ 。

$\text{mod\_exp}(x, y, n)$

1. if  $y = 0$  return(1);
2. if  $y \pmod 2 = 0$  return( $\text{mod\_exp}(x^2 \pmod n, y \div 2, n)$ );
3. return( $x \cdot \text{mod\_exp}(x^2 \pmod n, y \div 2, n) \pmod n$ )。

举一个例子,从 $\text{mod\_exp}(2, 21, 23)$ 开始,算法 4.3 将发出下面 5 次递归调用请求:

```

mod_exp(2, 21, 23)
= 2 · mod_exp(4(≡ 22 (mod 23)), 10, 23)           (第 3 步)
= 2 · mod_exp(16(≡ 42 (mod 23)), 5, 23)           (第 2 步)
= 2 · 16 · mod_exp(3(≡ 162 (mod 23)), 2, 23)       (第 3 步)
= 2 · 16 · mod_exp(9(≡ 32 (mod 23)), 1, 23)        (第 2 步)
= 2 · 16 · 9 · mod_exp(12(≡ 92 (mod 23)), 0, 23)   (第 3 步)
= 2 · 16 · 9 · 1                                   (第 1 步)

```

注意上面 6 行包含 $\text{mod\_exp}$ 的 5 次递归调用。最后一行“ $\text{mod\_exp}(12, 0, 23)$ ”只是表示“返回值 1”,并不是一个递归调用。最终返回给 $\text{mod\_exp}(12, 0, 23)$ 的值是 12,它是由第 3 步中的几个乘法产生的:

$$12 \equiv 2 \cdot 16 \cdot 9 \equiv 2^1 \cdot (2^2)^2 \cdot (((2^2)^2)^2) \pmod{23}$$

我们现在来考查算法 4.3 实现 $\text{mod\_exp}$ 的时间复杂度。对  $y > 0$ ,“除以 2”的运算恰好执行  $\lfloor \log_2 y \rfloor + 1$  次就得到商 0,运行 $\text{mod\_exp}(x, y, n)$ 要自身递归调用恰好  $\lfloor \log_2 y \rfloor + 1$  次,达到第 1 步中的终止条件(0 指数)。每一次递归调用包括一次平方或一次平方外加一次乘法,开销是  $O_B((\log x)^2)$ 。那么,假设  $x, y$  是小于  $n$  的数,算法 4.3 实现 $\text{mod\_exp}$ 的时间复杂度上界是  $O_B((\log x)^3)$ 。

与计算 gcd 时看来不可避免的相续性类似,计算 $\text{mod\_exp}$ 也有内在的相续性。这可以看成重复平方时的一个简单事实: $x^4$  只能在计算出了  $x^2$  后才能计算,依此类推。多年以来,改进其复杂度  $O_B((\log x)^3)$  并没有取得重大进展(不考虑使用 FFT,回顾 4.3.2.4 中的讨论)。

图 4.3 总结了基本模算术运算的时间复杂性。应注意到,对加和减,模运算不涉及除法;这是因为,对  $0 \leq a, b < n$ ,有  $-n < a \pm b < 2n$ ,因此

$$a \pm b \pmod n = \begin{cases} a \pm b & \text{如果 } 0 \leq a \pm b < n \\ a \pm b - n & \text{如果 } a \pm b \geq n \\ n + (a \pm b) & \text{如果 } a \pm b < 0 \end{cases}$$



运算 $a, b \in_U [1, n]$	复杂度
$a \pm b \pmod n$	$O_B(\log n)$
$a \cdot b \pmod n$	$O_B((\log n)^2)$
$b^{-1} \pmod n$	$O_B((\log n)^2)$
$a/b \pmod n$	$O_B((\log n)^2)$
$a^b \pmod n$	$O_B((\log n)^3)$

图 4.3 基本模算术运算按位计的时间复杂性

## 4.4 概率多项式时间

人们已经广泛接受,如果一种语言不属于 $\mathcal{P}$ ,那么不存在总能有效识别它的图灵机<sup>①</sup>。但是,有一类语言具有下列特性:没有证明它们属于 $\mathcal{P}$ ,但它们总能用一种图灵机有效地识别,有时候也会出错误。

这种机器有时候会出错的原因是有些操作步骤机器会做随机移动。尽管有些移动也产生正确的结果,而其他移动则会产生错误的结果。这种图灵机称为非确定性图灵机。这里要介绍的是判定性问题的一个子类,它具有下面的有界差错特性:

回答判定性问题时,非确定性图灵机出错概率的界是一个常数(概率空间是机器的随机纸带)。

习惯上称具有有界差错的非确定性图灵机为概率式图灵机。因此,“非确定性图灵机”实际上是指另一类不同的判定性问题,我们将在4.5节中介绍。

概率式图灵机也有多条纸带。其中有一条称为随机纸带,上面有一些均匀分布的随机字符。在扫描一个输入实例 $I$ 时,机器还将和随机纸带交互,读取一个随机字符,然后和确定性图灵机一样工作。该随机串称为概率式图灵机的随机输入。由于涉及到随机输入,概率式图灵机对输入实例 $I$ 的识别不再是 $I$ 的一个确定性函数,而与一个随机变量相关,也就是说机器随机输入的一个函数。该随机变量对识别 $I$ 造成了一定的差错概率。

概率式图灵机识别的语言类称为概率多项式时间(PPT)语言,用 $\mathcal{PP}$ 表示。

**定义 4.5  $\mathcal{PP}$ 类** 用 $\mathcal{PP}$ 表示具有下列特性的语言类。语言 $L$ 属于 $\mathcal{PP}$ ,如果存在一个概率式图灵机 $PM$ 和一个多项式 $p(n)$ ,使得对任意非负整数 $n$ , $PM$ 可以在时间 $T_{PM}(n) \leq p(n)$ 内以一定的差错概率识别任意实例 $I \in L$ ,其中差错概率是关于 $PM$ 随机移动的一个随机变量, $n$ 是表示实例 $I$ 规模的整数参数。

在定义4.5中,有一处含义特别模糊,那就是“ $PM$ 以一定的差错概率识别任意实例 $I \in L$ ”。“一定的差错概率”可以表示成以下两个条件概率界:

$$\text{Prob}[PM \text{ 识别 } I \in L | I \in L] \geq \epsilon \quad (4.4.1)$$

和

$$\text{Prob}[PM \text{ 识别 } I \in L | I \notin L] \leq \delta \quad (4.4.2)$$

其中 $\epsilon$ 和 $\delta$ 是参数,满足

<sup>①</sup> “有效机器”的精确含义在4.4.6节定义;这里我们粗略地说一个有效的机器就是一个很快的机器。

$$\epsilon \in \left(\frac{1}{2}, 1\right], \quad \delta \in \left[0, \frac{1}{2}\right) \quad (4.4.3)$$

概率空间是  $PM$  的随机纸带。

式(4.4.1)是正确识别一个实例的概率界,称为**概率(界)**。这里,“完全性”意味着最终识别语言中的一个实例。需要从下面界定这个概率是为了限制错误地拒绝一个实例的概率。下面的等价形式是式(4.4.1)更加明确的表示:

$$\text{Prob}[PM \text{ 识别 } I \notin L | I \in L] < 1 - \epsilon \quad (4.4.4)$$

在该式中,  $1 - \epsilon$  表示错误地拒绝的概率界。称  $PM$  的完全性是错误地拒绝的概率界。

式(4.4.2)是错误地识别一个非实例的概率界,称为**稳妥性概率(界)**,“稳妥性”表示没有将一个非实例识别为实例。很明显需要限制这个概率界。我们称  $PM$  的稳妥性是错误地识别非实例的概率界。

#### 4.4.1 差错概率的特征

我们用式(4.4.3)两个区间中的两个不精确常数  $\epsilon$  和  $\delta$  表示了  $PM$  的差错概率界,现在解释这种不精确不会带来任何问题。

##### 4.4.1.1 多项式时间特征

对概率式图灵机  $PM$ ,其差错概率界为确定值  $\epsilon \in (\frac{1}{2}, 1]$  (完全性)、 $\delta \in [0, \frac{1}{2})$  (稳妥性),对输入  $I$  重复运行  $n$  次  $PM$ ,表示为  $PM'(I, n)$ ,它仍然是一个概率多项式图灵机。我们可以用“大数判别”作为  $PM'(I, n)$  确定是接受还是拒绝  $I$  的准则。也就是说,如果  $\lfloor \frac{n}{2} \rfloor + 1$  或更多次  $PM(I)$  的运行都输出接受(拒绝),那么  $PM'(I, n)$  就接受(拒绝)。显然,  $PM'(I, n)$  的稳妥性和完全性概率是关于  $n$  的函数。现在证明  $PM'(I, n)$  仍然是关于  $I$  规模的多项式时间。

由于  $n$  次运行  $PM(I)$  的随机移动是相互独立的,  $PM(I)$  的每一次运行都可以看做是一个贝努利试验,“成功”的概率为  $\epsilon$  (或稳妥性  $\delta$ ),失败的概率为  $1 - \epsilon$  (或稳妥性  $1 - \delta$ )。由二项分布(见 3.5.2 节),大数判别准则给出  $PM'(I, n)$  的差错概率界是  $n$  次贝努利试验成功了  $\lfloor \frac{n}{2} \rfloor + 1$  次或更多次的概率之和。对于完全性,其和式为

$$\epsilon(n) = \text{Prob}\left[\xi_n \geq \left\lfloor \frac{n}{2} \right\rfloor + 1\right] = \sum_{i=\left\lfloor \frac{n}{2} \right\rfloor + 1}^n b(i; n, \epsilon) \quad (4.4.5)$$

对于稳妥性,我们有

$$\delta(n) = \text{Prob}\left[\eta_n \geq \left\lfloor \frac{n}{2} \right\rfloor + 1\right] = \sum_{j=\left\lfloor \frac{n}{2} \right\rfloor + 1}^n b(j; n, \delta) \quad (4.4.6)$$

这两个表达式都是各自二项分布的累加。因为  $\epsilon > \frac{1}{2}$  和  $\delta < \frac{1}{2}$ ,第一个分布的中心项(在 3.5.2 节中定义)在  $(n+1)\epsilon > \lfloor \frac{n}{2} \rfloor + 1$  点(二项式的项在该点达到最大值);另一个分布的中心项在  $(n+1)\delta < \lfloor \frac{n}{2} \rfloor + 1$  点。

在 3.5.2.1 节,我们讨论了这些和的特点。由于  $\lfloor \frac{n}{2} \rfloor + 1 > (n+1)\delta$ , 式(4.4.6)式中的和为二项分布的“右尾部”。应用式(3.5.7), 令  $r = \lfloor \frac{n+1}{2} \rfloor$  和  $p = \delta$ , 我们得到

$$\delta(n) < \frac{2(1-\delta)}{(1-2\delta)^2} \cdot \frac{1}{n+1}$$

由于  $\delta$  是一个常数, 则有

$$\delta(n) \rightarrow 0 \quad (n \rightarrow \infty)$$

读者可类似地推出下述结论

$$\epsilon(n) > 1 - \frac{c}{n}$$

其中  $c$  为某个常数。推导留做一个练习(见习题 4.7, 那里给出了一个提示)。

由于“尾部”比  $\frac{1}{n}$  趋于 0 的速度更快<sup>①</sup>, 我们可以令  $n = |I|$ , 则图灵机  $PM'(I, n)$  的运行时间在  $|I| \cdot \text{poly}(|I|)$  内, 其中  $\text{poly}(|I|)$  是  $PM$  机对实例  $I$  的运行时间。因此,  $PM'$  仍是多项式时间的。

#### 4.4.1.2 为什么界要偏离 $\frac{1}{2}$

如果  $\epsilon = \delta = \frac{1}{2}$ , 那么两个分布(4.4.5)和(4.4.6)的中心项都是点  $\lfloor \frac{n}{2} \rfloor$ 。容易检验, 对奇数  $n$  有

$$\epsilon(n) = \delta(n) = \frac{1}{2}$$

对偶数  $n$  有

$$\epsilon(n) \approx \delta(n) \approx \frac{1}{2}$$

也就是说, 不管重复运行  $PM(I)$  多少次,  $\epsilon(n)$  不可能放大, 而  $\delta(n)$  也不可能缩小, 它们将保持在  $\frac{1}{2}$  水平。因此  $PM'(I, n)$  即使独立运行  $n$  次也不能作出决定, 因为无论是稳妥性还是完全性, 运行  $n$  的一半次数将接受, 另一半将拒绝。由于  $n$  没有限制,  $PM(I)$  一直处在不能确定的状态, 图灵机  $PM'(I, n)$  永远不能终止, 因此不可能是一个多项式时间算法。

因此, 由于  $\mathcal{PP}$  是成员资格可以在概率多项式时间内识别的语言类, 我们必须要求式(4.4.1)和式(4.4.2)中的差错概率界要偏离  $\frac{1}{2}$ 。

但是应该注意到, 对必然包括“双边差错”问题子类(见 4.4.5 节)的  $\mathcal{PP}$  类中最一般的语言识别问题, 差错概率界要偏离  $\frac{1}{2}$  只是一个必要条件。如果一个问题“单边差错”的(即或者  $\epsilon = 1$ , 或者  $\delta = 0$ , 见 4.4.3 节和 4.4.5 节), 那么差错概率界偏离  $\frac{1}{2}$  是不必要的。这是因为, 对于单边差错算法, 我们不必使用大数判别准则, 可以用“少数判别准则”代替。例如,  $PM'(I, n)$  可以使用“一致判别准则”, 仅当运行  $n$  次  $PM(I)$  得到同样的结果才识别或拒绝  $I$ 。在这样的判别准则里, 对任意  $\epsilon, \delta \in (0, 1)$ , 以指数的速度有  $\epsilon(n) \rightarrow 1$  或  $\delta(n) \rightarrow 0$ 。

<sup>①</sup> 在式(3.5.7)和式(3.5.8)中导出的只是两个上界, 尾部趋于 0 的实际速度远远比  $\frac{1}{n}$  快。参看例 3.9 中的数字。这得到了 18.5.5.11 节中协议 18.4 完全性和稳妥性的进一步证实。

在应用中,有可能一些有用的问题具有  $\epsilon \leq \frac{1}{2}$  或  $\delta \geq \frac{1}{2}$  (但是,正如我们所推导的一样,不必要求二者都成立)。对这样的问题,改变判别准则(如少数判别准则)可能给我们提供放大或缩小差错概率的机会。在 18.5.1 节将看到一个协议例子,其识别概率是  $\epsilon = \frac{1}{2}$ ,但是用少数判别准则,通过重复协议仍然能够放大完全性概率。

### 几个 $PP$ 子类

有几个  $PP$  中的子类,分别使用不同的  $\epsilon$  和  $\delta$  值,由式(4.4.1)和式(4.4.2)中的差错概率界表达式给出定义。现在来介绍这几个子类,对每一个子类用算法给出一个例子。与确定性图灵机仿真多项式时间算法类似,概率式图灵机也可以仿真随机化(多项式时间)算法。因此,介绍中给出的算法例子并不局限于语言识别。

#### 4.4.2 “总是快速且正确的”子类

$PP$  的一个子类称为  $ZPP$  (它表示零-边差错概率多项式时间),如果式(4.4.1)和式(4.4.2)中的差错概率界具有下列特性:对任意  $L \in ZPP$ ,存在一个随机算法  $A$  对任意实例  $I$  满足

$$\text{Prob}[A \text{ 识别 } I | I \in L] = 1$$

和

$$\text{Prob}[A \text{ 识别 } I | I \notin L] = 0$$

这一概率特性表示随机化算法里的随机操作根本就没有出错。因此,初一看, $ZPP$  应该和  $P$  没有区别。但是,有一类问题它既可以用确定性算法求解,也可以用随机化算法求解,都是多项式时间。尽管所用的随机化算法不会产生任何差错,但比对应的确定性算法快得多。稍后我们给出一个例子来比较它们的时间复杂性。

##### 4.4.2.1 一个“零-边差错”算法例子

有的随机化算法非常自然,以至于我们使用它们已经很长时间而不用相应的确定性算法。例如,为了用砵秤<sup>①</sup>称一个物体,使用者用随机的方式在秤杆上移动秤砣,称出物体的重量,这比用确定性方式快得多。我们熟悉的一个类似算法是从电话簿中查询某人电话号码的随机化过程。该算法在算法 4.4 中描述。

显然,算法 4.4 中的随机操作不会给输出结果带来任何差错,因此,这事实上就是一个“零-边差错”随机化算法。对于一个有  $N$  页的电话簿,算法 4.4 只需执行  $O(\log N)$  步就可以找到包含某人姓名和电话号码的那一页。我们应当注意,确定性算法“查电话簿”平均要执行  $O(N)$  步。

算法 4.4 工作得这么快的原因在于电话簿中的姓名已经按字母排序了。应当注意到,排序本身就是一个  $ZPP$  问题:“快速分类法”(参阅[9]的 92~97 页)就是一个随机化算法,可以在  $O(N \log N)$  步内对  $N$  个元素归类,它的随机操作也不会给输出结果带来任何差错。相反,“冒泡-分类法”是一个确定性分类算法,对  $N$  个元素分类要  $O(N^2)$  步(参阅[9]的 77 页)。

<sup>①</sup> 这种称重工具在中文里叫杆秤,已经使用 2000 多年。

**算法 4.4 查电话簿(ZPP算法)**

输入 姓名:某人的名字;

登记簿:电话簿;

输出 此人的电话号码。

1. 重复下列步骤直到电话簿(*Book*)只有1页

{

(a) 随机翻开电话簿的一页;

(b) 如果人名在该页前面,  $Book \leftarrow \text{Earlier\_pages}(Book)$ ;

(c) 否则  $Book \leftarrow \text{Later\_pages}(Book)$ ;

}

2. Return(人名边的电话号码);

我们可以说,ZPP是用随机化算法“总能快速正确”识别的语言的一个子类。

**4.4.3 “总是快速且很可能正确的”子类**

PP的一个子类称为PP(Monte Carlo) (“Monte Carlo”通常被典型地用做“随机化”的通用术语),如果式(4.4.1)和式(4.4.2)中的差错概率界具有下列特性:对任意  $L \in \text{PP}(\text{Monte Carlo})$ , 存在一个随机化算法 *A*,使得对任意实例 *I*,有

$$\text{Prob}[A \text{ 识别 } I \mid I \in L] = 1$$

且

$$\text{Prob}[A \text{ 识别 } I \mid I \notin L] \leq \delta$$

这里, $\delta$ 是在区间 $(0, \frac{1}{2})$ 内的任意常数。但正如我们在4.4.1.2节指出的,对于单边差错算法,在缩小稳妥性差错概率界的过程中,由于不必使用大数判别准则, $\delta$ 实际上可以是属于 $(0,1)$ 的任何常数。

注意,现在 $\delta \neq 0$ ,否则,该子类退化为特例ZPP。具有这种差错概率特征的随机化算法在稳妥性方面有单边差错。换句话说,这样的算法有可能出错,错误地接受一个非实例。但是,如果输入的确是一个实例,那么,它总是会被识别。算法的这个子类称为**Monte Carlo 算法**。

根据在4.4.1节中的学习,我们知道重复的迭代 Monte Carlo 算法仍然是多项式时间的,通过独立重复 Monte Carlo 算法,它的差错概率可以任意接近0。因此我们说 Monte Carlo 算法总是很快,而且很可能是正确的。

现在证明 PRIMES(全体素数的集合)属于PP(Monte Carlo)子类。

**4.4.3.1 一个 Monte Carlo 算法例子**

根据费马定理,显然,如果  $p$  是一个素数,  $x$  与  $p$  互素,那么  $x^{p-1} \equiv 1 \pmod{p}$ 。这构成了下面 Monte Carlo 方法素性检测([284])的基础,也就是说,选取  $x \in_U(1, p-1]$ ,  $\text{gcd}(x, p) = 1$ , 检验

$$x^{(p-1)/2} \stackrel{?}{\equiv} \pm 1 \pmod{p} \quad (4.4.7)$$

重复检测  $k = \log_2 p$  次, 并且  $-1$  至少出现一次。算法 4.5 描述了这一检测算法。

#### 算法 4.5 概率素性检测 (Monte Carlo 算法)

输入  $p$ : 一个正整数;

输出 如果  $p$  是素数, 输出 YES; 否则输出 NO。

Prime\_Test( $p$ )

1. 重复  $\log_2 p$  次:

(a)  $x \in_U(1, p-1]$ ;

(b) 如果  $\gcd(x, p) > 1$  或  $x^{(p-1)/2} \not\equiv \pm 1 \pmod{p}$  return(NO);

结束重复;

2. 如果 1.(b) 中的检测从没有出现  $-1$ , return(NO);

3. return(YES)。

首先, 根据费马小定理(6.4 节的定理 6.10), 我们知道, 如果  $p$  是素数, 那么对所有的  $x < p$  有

$$x^{p-1} \equiv 1 \pmod{p} \quad (4.4.8)$$

因此, 如果  $p$  是素数, 那么 Prime\_Test( $p$ ) 总是返回 YES, 也就是说, 我们总有 (包括  $p$  是偶素数的情形)

$$\text{Prob}[x^{(p-1)/2} \equiv \pm 1 \pmod{p} | p \text{ 是素数}] = 1$$

另一方面, 如果  $p$  是一个合数, 那么同余式(4.4.7)一般情况下不会成立。事实上(群论中的一个事实, 见习题 5.2.3 和 5.2.1 节中的定理 5.1), 如果同余式(4.4.7)对某个  $x < p$ ,  $\gcd(x, p) = 1$  不成立, 那么该不等式表明至少有一半的数属于这一类。于是我们得到对  $x \in_U(1, p-1]$ ,  $\gcd(x, p) = 1$ , 有

$$\text{Prob}[x^{(p-1)/2} \equiv \pm 1 \pmod{p} | p \text{ 是合数}] \leq 1/2 \quad (4.4.9)$$

因此, 如果对均匀随机选取的  $x$  通过了  $k$  次 (记住  $-1$  的情形至少发生一次) 检测, 那么  $p$  不是素数的概率不大于  $2^{-k}$ 。这里我们使用“一致判别准则”: 如果在  $\log_2 p$  次检测中只要有一次失败, 这个  $p$  就被拒绝。注意这一判别准则与大数判别准则不同, 在 4.4.1 节讨论的大数判别准则中 (双边差错概率问题的一般情形), 失败是可以容忍的, 只要失败的检测次数不超过半数。在这里的“一致判别”中, 稳妥性概率趋于 0 的速度比大数判别的情形快得多。

我们已经令  $k = \log_2 p$ , 因此对任意输入实例  $p$  有

$$\text{Prob}[\text{Prime\_Test}(p) = \text{YES} | p \text{ 不是素数}] \leq 2^{-\log_2 p}$$

在 4.3 节中已经看到, 对  $\log_2 p$  比特长的输入, 计算模指数和最大公因子的时间复杂度界都是  $O_B((\log_2 p)^3)$ 。因此, Prime\_Test( $p$ ) 的时间复杂度的界是  $O_B((\log p)^4)$ 。

到这里我们知道全体素数的语言 PRIMES 属于  $\mathcal{PP}$  (Monte Carlo)。

不过, 尽管没有使这一论断失效, 2002 年 8 月三位印度计算机科学家 Agrawal、Kayal 和 Saena 发现了一种确定性多项式时间素性检测算法[8]; 因此, PRIMES 事实上属于  $\mathcal{P}$ 。

#### 4.4.4 “很可能快且总是正确的”子类

称  $\mathcal{PP}$  的一个子类为  $\mathcal{PP}(\text{Las Vegas})$  (以美国著名赌城 Las Vegas 命名), 如果式(4.4.1)和



式(4.4.2)中的差错概率界具有下列特征:对任意  $L \in \mathcal{PP}(\text{Las Vegas})$ , 存在一个随机化算法  $A$ , 使得对任意实例  $I$  有

$$\text{Prob}[A \text{ 识别 } I | I \in L] \geq \epsilon$$

且

$$\text{Prob}[A \text{ 识别 } I | I \notin L] = 0$$

这里,  $\epsilon$  是区间  $(\frac{1}{2}, 1)$  内的任意常数。又如对于 4.4.3 节中稳妥性的单边差错, 在放大完全性差错概率界的过程中, 因为我们不必使用大数判别准则, 所以  $\epsilon$  实际上可以是  $(0, 1)$  内任意常数。

我们还应该注意到  $\epsilon \neq 1$ , 否则该子类退化为  $\mathcal{ZPP}$  的一种特例。具有这种差错概率特征的随机化算法在完全性方面有单边差错。换句话说, 这种算法有可能出错, 错误地未识别到一个实例。但是, 如果识别了一个实例, 那么它不可能是错的。算法的这个子类称为 **Las Vegas 算法**。Las Vegas 一词首先在 [16] 中引入, 指要么给出正确答案, 要么不给出任何答案的随机化算法。

根据 4.4.1.1 节的分析, 我们知道, 迭代 Las Vegas 算法仍然是多项式时间的, 通过独立重复 Las Vegas 算法, 它对一个实例回答 YES 的概率可以加大到任意接近于 1。因此, 如果我们说 Monte Carlo 算法总是很快的, 而且很可能是正确的, 那么 Las Vegas 算法就总是正确的, 而且很可能是很快的。

观察  $\mathcal{ZPP}$ 、 $\mathcal{PP}(\text{Monte Carlo})$  和  $\mathcal{PP}(\text{Las Vegas})$  的概率特征, 下列等式是显然的:

$$\mathcal{ZPP} = \mathcal{PP}(\text{Monte Carlo}) \cap \mathcal{PP}(\text{Las Vegas})$$

#### 4.4.4.1 一个 Las Vegas 算法例子

设  $p$  是一个正奇数,  $p-1 = q_1 q_2 \dots q_k$  是  $p-1$  的完全整数分解(某些素因子可能重复)。在第 5 章将给出一个事实(5.4.4 节的定理 5.12):  $p$  是素数的条件是当且仅当存在一个正整数  $g \in [2, p-1]$ , 使得

$$\begin{aligned} g^{p-1} &\equiv 1 \pmod{p} \\ g^{(p-1)/q_i} &\not\equiv 1 \pmod{p}, i = 1, 2, \dots, k \end{aligned} \quad (4.4.10)$$

这个事实给出了一个素性证明的算法。输入一个奇素数  $p$  和  $p-1$  的完全分解, 该算法试图寻找一个  $g$  满足式(4.4.10)。如果找到了这种数, 算法输出 YES 并成功地终止, 且  $p$  必是素数。否则, 算法将处于不确定状态, 也就是说, 它不知道  $p$  是否是一个素数。该算法在算法 4.6 中描述。

由于  $k \leq \log_2(p-1)$ , 因此算法 4.6 在关于  $p$  的规模的多项式时间内结束。

---

#### 算法 4.6 素性证明(Las Vegas 算法)

输入  $p$ : 一个正奇数;

$q_1, q_2, \dots, q_k$ :  $p-1$  的全体素因子;

输出 如果  $p$  是素数, 输出 YES, 否则输出 NO;

以一定的错误概率输出 NO\_DECISION。

1. 选取  $g \in_{\mathcal{U}} [2, p-1]$ ;

2. for( $i = 1, i++ , k$ ) do

- 如果  $g^{(p-1)/q_i} \equiv 1 \pmod{p}$ , 输出 NO\_DECISION 并中止程序;
3. 如果  $g^{(p-1)} \not\equiv 1 \pmod{p}$ , 输出 NO 并中止程序;
4. 输出 YES 并终止程序。

根据 5.4.4 节的定理 5.12, 我们将看到, 如果算法 4.6 输出 YES, 那么输入整数必定为素数, 不可能出错; 如果算法输出 NO, 回答也是正确的, 因为否则费马小定理(见 4.4.8 节)将失效。这两种情况表明算法“总是正确”的特性。算法无差错的特性使它可以称为“素性证明。”

不过, 如果算法 4.6 输出 NO\_DECISION, 就不知道输入  $p$  是不是一个素数了。有可能不是一个素数, 但也有可能出错。对于后者,  $p$  实际上是一个素数, 只不过算法随机选取用于检验的数  $g$  有误。学习 5.4.4 节中定理 5.12 后, 我们会明白出错的数  $g$  不是一个“本原根”。

到这里已经知道, 算法 4.6 在完全性方面是一个单边差错算法, 即 Las Vegas 算法。可以将该算法修改成在应答 NO\_DECISION 时不终止, 而是另外选取一个测试数  $g$  执行测试。修改后的算法仍然是一个 Las Vegas 算法, 并且“很可能快”, 因为有可能总是选到非本原根作为测试数。幸运的是, 对任意奇数  $p$ , 模  $p$  乘法群(将在第 5 章定义)包含许多本原根, 对模  $p$  群随机抽样, 会以不可忽略的概率抽到这种元素(第 5 章将给出模素数的乘群中本原根的比例)。

Las Vegas 算法和 Monte Carlo 算法统称为“单边差错的随机化算法”。这些算法(回忆一下, 它包括 ZPP)事实上是有效的, 尽管是非确定性算法, 它们在时间复杂度方面和 P 中的算法类似。

#### 4.4.4.2 另一个 Las Vegas 算法例子: 量子整数分解

量子计算机可以在关于整数长度的多项式时间内分解整数(即 FACTORIZATION  $\in$  QP)。Schor 设计了一个这类算法([269], 也可参阅[302]的 108 ~ 115 页)。现在解释 Schor 的量子分解算法也是一个 Las Vegas 算法。

要分解整数  $N$ , 选取一个随机整数  $a$ , 量子算法可以确定函数  $f(x) = a^x \pmod{N}$  的周期, 即找到满足  $f(r) = 1$  的最小整数  $r$ , 其中该量子算法从傅里叶变换[278]取样, 用 Simon 的方法确定量子状态的周期。在第 6 章将看到, 对合数  $N$ , 存在整数  $a$  满足  $\gcd(a, N) = 1$  且有偶周期(称为元素  $a$  的乘法阶), 即  $r$  是一个偶数。

一旦找到偶周期  $r$ , 如果  $a^{r/2} \not\equiv \pm 1 \pmod{N}$ , 那么  $a^{r/2} \pmod{N}$  是 1 模  $N$  的一个非平凡平方根。在 6.6.2 节(定理 6.17)我们将证明  $\gcd(a^{r/2} \pm 1, N)$  必定是  $N$  的一个非平凡因子, 即该算法成功地分解了  $N$ 。

如果  $a^{r/2} \equiv \pm 1 \pmod{N}$ , 那么  $\gcd(a^{r/2} \pm 1, N)$  是  $N$  的平凡因子, 即 1 或  $N$ , 此时算法失败, 不能给出答案。但是, 对于随机选取的整数  $a < N$ , 出现  $a^{r/2} \equiv \pm 1 \pmod{N}$  的概率下界为一个常数  $\epsilon > 1/2$ , 因此, 该算法可重新使用另一个随机整数  $a$ 。根据在 4.1.1 节中的分析, Schor 算法仍然是多项式时间的。

#### 4.4.5 “很可能快且很可能正确的”子类

PP 的一个子类称为 BPP(它表示“有界差错概率多项式时间”), 条件是式(4.4.1)和式(4.4.2)中的差错概率界对下述情形都成立:

$$\epsilon \in \left[\frac{1}{2} + \alpha, 1\right) \text{ 和 } \delta \in \left(0, \frac{1}{2} - \beta\right] \quad (4.4.11)$$

这里  $\alpha > 0$  且  $\beta > 0$ 。对该差错概率特性,我们应该注意以下两点:

1.  $\epsilon \neq 1$  且  $\delta \neq 0$ 。否则,子类  $\mathcal{BPP}$  退化为下面三种更简单情形的某一种:  $\mathcal{ZPP}$ , 或者  $\mathcal{PP}$  (Monte Carlo), 或者  $\mathcal{PP}$  (Las Vegas)。现在由于  $\epsilon \neq 1$  且  $\delta \neq 0$ ,  $\mathcal{BPP}$  中的算法具有双边差错, 错误地拒绝(完全性差错)和错误地识别(稳妥性差错)都有可能。
2.  $\alpha > 0$  且/或  $\beta > 0$ 。这显然意味着  $\mathcal{BPP}$  中算法的差错概率界偏离  $\frac{1}{2}$ 。我们在 4.4.1 节已经得出, 如果  $\epsilon \neq \frac{1}{2}$  ( $\delta \neq \frac{1}{2}$ ), 那么使用大数判别准则, 重复该算法可以放大完全性(缩小稳妥性)差错概率。如果  $\epsilon = \frac{1}{2}$  或  $\delta = \frac{1}{2}$ , 那么就不能用大数判别准则, 因为前者(后者)意味着不存在多数随机移动得出一个识别(拒绝)。但是, “少数判别准则”还可以用(在 18.5.1 节将看到一个这样的例子)。最后, 如果  $\epsilon = \frac{1}{2}$  且  $\delta = \frac{1}{2}$ , 那么, 不能用任何判别准则, 该问题不属于  $\mathcal{PP}$  (即不论运行多长时间都不能用一个非确定性图灵机识别)。

除 Monte Carlo 和 Las Vegas 外, Atlantic City 是另外一个著名赌博地点, 通过增加赌博游戏种类来提高赢的概率, 以此来诱惑人们, 所以具有双边差错的随机化算法也称为 **Atlantic City 算法**。现在来看一个 Atlantic City 算法的例子。

#### 4.4.5.1 一个 Atlantic City 算法例子

在量子密码学里有一个著名的协议称为量子密钥分配协议(QKD 协议, 见[32])。QKD 协议可以在两个通信实体之间协商一个比特串, 无须两方会面, 而且双方可以高置信度地确信协商的比特串除了他们之外没有第三方知道。QKD 协议是一个双边差错概率算法。下面描述该算法并检验它的双边差错概率特性。

首先对 QKD 协议的物理原理做一个简单的描述。在 QKD 协议里, 一个秘密比特串的分配是通过发送者(设 Alice 是发送者)传输一个 4 极化方式光子来实现的。每个光子都处于某个状态(称为光子态或状态), 用下面的四个符号表示:

$$—, |, /, \backslash$$

前两个光子状态用设置为垂直/水平方向的极化器发射, 后两个光子状态用设置为对角方向的极化器发射, 分别用 + 和 × 表示这两个不同方向的极化器。我们可以将信息编码为这四个光子状态。下面是一个比特到光子状态的编码方案:

$$+(0) = —, +(1) = |, \times(0) = /, \times(1) = \backslash \quad (4.4.12)$$

这一编码方案是公开的。如果 Alice 想传输一个传统比特 0(1), 她可以选择用 + 在量子信道上发送 —(|); 或选择用 × 发送 /(\)。对 QKD 协议传输的每一个传统比特, Alice 都要均匀随机地设置不同方向的极化器 + 或 ×。

为了接收一个量子状态, 接收者(可以是意定接收者 Bob 或窃听者 Eve)需要使用一个叫量子观测仪的设备, 它也要设置成垂直/水平方向或对角方向。我们仍分别用 + 和 × 表示这两个不同方向的检测仪。设  $\xrightarrow{+}$  和  $\xrightarrow{\times}$  表示两个不同方向的观测仪接收和截获从左到右传输的量子状态。光子状态的检测遵循以下规则:

正确检测(以概率 1 保持光子状态)

$$- \xrightarrow{+} -, \quad | \xrightarrow{+} |, \quad / \xrightarrow{\times} /, \quad \backslash \xrightarrow{\times} \backslash$$

错误检测(光子状态被破坏)

$$\begin{array}{l} / \xrightarrow{+} - \quad \begin{array}{l} \text{概率 } \frac{1}{2} \\ \text{概率 } \frac{1}{2} \end{array}, \quad \backslash \xrightarrow{+} - \quad \begin{array}{l} \text{概率 } \frac{1}{2} \\ \text{概率 } \frac{1}{2} \end{array} \\ - \xrightarrow{\times} / \quad \begin{array}{l} \text{概率 } \frac{1}{2} \\ \text{概率 } \frac{1}{2} \end{array}, \quad | \xrightarrow{\times} / \quad \begin{array}{l} \text{概率 } \frac{1}{2} \\ \text{概率 } \frac{1}{2} \end{array} \end{array}$$

这些检测规则说明了下面的事实。通过正确设置监测仪为垂直/水平方向,垂直/水平方向的量子状态可以正确地检测到;类似地,通过正确设置监测仪为对角方向,对角方向的量子状态也可以正确地检测到。但是,如果垂直/水平(对角)方向的量子状态用一个对角(垂直/水平)方向的监测仪检测,那么会出现  $\pm 45^\circ$  的方向偏转,为任何一个方向的概率都是 0.5,会发生错误的检测,这是“Heisenberg 测不准原理”的必然结果,也是 QKD 协议的工作原理。

因此,如果接收者的监测仪设置的方向和 Alice 的极化器设置的一致(也就是相同),就可以正确接收光子状态。式(4.4.12)中公开的比特-光子编码方案是传统比特和光子状态之间的 1-1 映射。因此,在这种情况下,Alice 发送的传统比特可以正确地解码。另一方面,如果两端光子设备设置的方向不一样,必然出现检测错误,并且必然破坏传输的光子状态,尽管接收者并不知道实际上已经发送和破坏了哪些光子状态。

现在就可以描述 QKD 协议了。该协议在协议 4.1 中给出。

#### 协议 4.1 量子密钥分配(Atlantic City 算法)

##### 协议的高层描述

**量子信道** Alice 发送  $m$  个量子状态给 Bob,每一个都在  $\{-, |, /, \backslash\}$  中随机选取方向。

**传统信道,公开会话** 作为 Alice 的极化器和 Bob 的观测仪设置一致的结果,他们选出  $k = \frac{m}{10}$  个传输的“筛选比特”。在  $k$  个筛选比特中,进一步随机比较  $\ell (< k)$  个“测试比特”,检测窃听。如果没有窃听者存在,他们对余下的  $k - \ell$  个秘密比特就达成了一致。

1. Alice 生成  $m$  个传统随机比特  $a_1, a_2, \dots, a_m \in_U \{0, 1\}$ ; 随机设置  $m$  个极化器  $p_1, p_2, \dots, p_m \in_U \{+, \times\}$ ; 按照式(4.4.12)中的比特到光子的编码方案,向 Bob 发送  $m$  个光子状态  $p_1(a_1), p_2(a_2), \dots, p_m(a_m)$ ;
2. Bob 随机地设置  $m$  个光子观测仪  $o_1, o_2, \dots, o_m \in_U \{+, \times\}$  并用它们接收  $m$  个光子状态; 使用式(4.4.12)中的比特到光子的编码方案解码得到传统比特  $b_1, b_2, \dots, b_m$ ; 告诉 Alice: “都收到了!”;
3. 公开比较他们的设置  $(p_1, o_1), (p_2, o_2), \dots, (p_m, o_m)$ ; 如果超过  $k = \frac{m}{10}$  对设置一致: (\* 不失一般性,我们可以重新给出这些比特的下标 \*)

$$p_i = o_i, \quad 1 \leq i \leq k$$

- 那么他们继续执行下面的步骤,否则运行失败(\*这一失败是完备性完全性差错\*);
4. (\*现在集合 $\{(a_i, b_i)\}_{i=1}^k$ 中包含了通过极化器和检测仪一致设置分配的 $k$ 对“筛选比特”\*) Alice 和 Bob 随机地公开比较 $\{a_i, b_i\}_{i=1}^k$ 中的 $l$ 对;比较的比特称为测试比特;如果测试比特中有任何一对比特不一致,他们宣布“检测到窃听者!”并放弃本次运行;
  5. 输出剩下的 $k-l$ 比特作为分配的密钥;协议成功终止(\*但可能发生了稳妥性差错\*).

我们解释一下协议是如何工作的,并计算发生双边差错的概率。

第1、2步很直接:使用 $m$ 个随机设置 $p_1, p_2, \dots, p_m \in_U \{+, \times\}$ , Alice 发送给 Bob  $m$ 个随机光子状态(第1步), Bob 不得不用 $m$ 个随机设置 $o_1, o_2, \dots, o_m \in_U \{+, \times\}$ 来检测(第2步)。Alice 编码和发送的 $m$ 个传统比特是 $a_1, a_2, \dots, a_m$ , Bob 接收和解码得到的是 $b_1, b_2, \dots, b_m$ 。

在第3步, Alice 和 Bob 在传统通信信道上进行讨论,看他们的 $m$ 个设备随机设置对 $\{(p_i, o_i)\}_{i=1}^m$ 中是否有 $k = \frac{m}{10}$ 对设置是一样的。如果有 $k$ 对设置是一样的,他们将继续协议;否则出现完全性差错,协议失败。稍后我们将给出完全性差错的概率度量。

假设没有发生完全性差错,双方现在执行第4步。他们已经有了通过 $k$ 次一致的设备设置分配的 $k$ 个筛选比特。不失一般性,可以重新标记这些比特的下标。这样, Alice 的筛选比特是 $a_1, a_2, \dots, a_k$ , Bob 的筛选比特是 $b_1, b_2, \dots, b_k$ 。他们现在在传统信道上进行公开的讨论,随机比较 $l$ 对筛选比特。任何不一致都将视为是由窃听者 Eve 引起的。如果他们在第4步没有发现 Eve 的存在,协议在第5步愉快地结束。Alice 和 Bob 现在共享了认为没有被窃听的 $k-l$ 比特。但是,有可能没有检测到窃听是因为发生了稳妥性差错。现在来讨论这种差错的概率。

### 稳妥性差错概率

假设 Eve 侦听了量子信道。Eve 检测 Alice 发送的光子状态的惟一方法是用 Bob 采用的同样技术。因此 Eve 不得不设置 $m$ 个随机方向进行观测,她还必须发送 $m$ 个状态给 Bob。由“Heisenberg 测不准原理”,她的错误观测会破坏 Alice 发送的光子状态。既然 Eve 不知道她的观测是否正确,她也不会知道该发送什么状态给 Bob。Eve 的一个策略是向 Bob 发送她随机产生的 $m$ 个全新状态(就像 Alice 发送的一样),希望无论她和 Alice 发送什么 Bob 都检测不到差别;另外一个策略就是无论她检测到什么她都只是转发给 Bob,希望她没有破坏 Alice 发送的光子状态。实际上,对我们正要推导的稳妥性差错概率的影响来说,这两种策略没有什么分别。

考虑第二个策略(第一个策略将导致同样的稳妥性差错概率结果,见习题4.9)。对状态 $p_i(a_i)$ ,如果 Eve 正确地设置了她的检测仪 $e_i$ ,即 $e_i = p_i$ ,那么,她会正确地接收到状态 $p_i(a_i)$ 和比特 $a_i$ ,然后 Bob 也会正确地收到该状态和比特。因此在这种情况下, Alice 和 Bob 无法检测到 Eve 的存在。由于 Eve 正确地设置她的第 $i$ 个检测仪的概率是 $1/2$ ,我们得到(在第 $i$ 个位置)没有检测到窃听的概率为 $1/2$ 。

如果 Eve 对她第 $i$ 个观测仪的设置不正确,那么观测的第 $i$ 个状态不正确,因此她将发送给 Bob 一个错误的状态。但是, Bob 将由于 $\pm 45^\circ$ 偏转“纠正”错误状态,每一个方向都是50:50

的机会。因此, Bob 接收到的状态或者正确或者错误, 每一种情况的概率都是  $1/2$ 。正确的接收又将导致检测不到 Eve。注意没检测到的情况发生在 Eve 错误地设置了她的设备之后, Eve 的错误设置概率也是  $1/2$ 。由于 Eve 和 Bob 的设备设置是相互独立的, 这种窃听没检测到的概率是  $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ 。

对上面两段得到的概率求和, 我们得到没检测到 Eve 窃听第  $i$  个状态的概率是  $\frac{1}{2} + \frac{1}{4} = \frac{3}{4}$ 。由于 Eve 为了得到分配的密钥必须侦听所有的筛选比特, Alice 和 Bob 要比较  $\ell$  个随机测试比特, 任何一个测试比特不一致都会发出检测到窃听的信号(这是一个“一致判别准则”, 即使一个错误也是不能容忍的, 见 4.4.1.2 节), Eve 在所有位置都未被检测到的概率是  $(\frac{3}{4})^\ell$ 。这就是稳妥性差错概率, 它很快就趋于 0。

### 完全性差错概率

最后我们来考察完全性差错发生的概率。假设 Alice 对她设备的  $m$  个设置是一个二进制向量  $V = (v_1, v_2, \dots, v_m)$ , Bob 的是  $W = (w_1, w_2, \dots, w_m)$ 。当

$$V \oplus W = (v_1 \oplus w_1, v_2 \oplus w_2, \dots, v_m \oplus w_m)$$

中少于  $\frac{m}{10}$  个 0 时, 会发生完全性差错。由于 Alice 和 Bob 的设置是独立均匀分布的,  $V \oplus W$  必定是  $m$  个二进制比特构成的一个均匀随机二进制向量。向量中 0 的个数  $i$  的概率服从  $m$  次试验有  $i$  次成功的二项分布, 其中每次成功的概率是 0.5。显然, 在向量  $V \oplus W$  中, “0 的最可能个数”是  $\frac{m}{2}$ 。也就是说, 这个二项分布的“中心项”(见 3.5.2.1 节)是在点  $\lfloor \frac{m+1}{2} \rfloor$ 。因此点  $\frac{m}{10}$  远远地偏离(远远地在左边)中心项所在的点  $\lfloor \frac{m+1}{2} \rfloor$ 。因此, 完全性差错概率

$$\text{Prob}[\text{zeros\_in}(V \oplus W) < \frac{m}{10}]$$

是这个二项分布函数的“左尾部”。根据在式(3.5.8)给出的“左尾部”概率界, 我们得到发生完全性差错的下述概率界

$$\text{Prob}[\text{zeros\_in}(V \oplus W) < \frac{m}{10}] < \frac{\left(m + 1 - \frac{m}{10}\right) 0.5}{\left((m + 1) 0.5 - \frac{m}{10}\right)^2} < \frac{3}{m}, \quad m \geq 2$$

因此, Alice 和 Bob 执行协议超过第 3 步的概率大于  $1 - \frac{3}{m}$ 。

### 双边差错概率小结

我们总结协议 4.1 的双边差错概率如下。对于完全性我们有

$$\text{Prob}[\text{筛选比特数} \geq \frac{m}{10} \mid \text{分配 } m \text{ 个量子状态}] > 1 - \frac{3}{m}$$

对稳妥性我们有

$$\text{Prob}[\text{没检测到 Eve} \mid \text{Alice 和 Bob 测试 } \ell \text{ 个比特}] \leq \left(\frac{3}{4}\right)^\ell$$



应当注意到,由式(3.5.8)得到的“左尾部”概率界 $\frac{3}{m}$ 对完全性差错概率只是一个不紧致的上界。左侧尾部趋于0的速度要比 $\frac{3}{m}$ 快得多(参阅例3.9中的数字)。

这些差错概率结果表明,QKD协议对密钥分配是实用的。商用QKD系统有望在2004年左右投入实际使用[270]。

在实际应用中,Alice和Bob进行公开讨论的传统信道要有认证功能。为了使他们确信在和正确的通信方共享密钥,这是必要的。认证是第IV部分的主题。

#### 4.4.6 有效算法

作为介绍多项式时间类和概率多项式时间(PPT)子类的结束,我们已经确立了类之间的包含关系如下:

$$P \subseteq ZPP \subseteq \begin{matrix} PP(\text{Monte Carlo}) \\ PP(\text{Las Vegas}) \end{matrix} \subseteq BPP \subseteq PP$$

能够解决以上任何一个类中问题的算法称为有效算法。

**定义 4.6 有效算法** 一个算法称为是有效的,在于它是确定的或随机化的,其运行时间可以表示为输入规模的多项式。

这一定义给出了**易处理性**(tractability)概念:不管是确定性的还是随机化的,多项式时间问题是易处理的,也就是说,即使这类问题的规模非常大,它要求的资源也是可以处理的。易处理问题类以外的问题是**难处理的**(intractable)。

但是,由于多项式的次数可能相差很大, $P$ 和 $PP$ 问题的时间复杂度也会有很大的不同。因此,求解可解问题的有效算法在实用的意义上不必是高效的。在本章后面将看到几个协议例子,其时间复杂度的界是输入规模的多项式。因此,这些协议是有效的(根据定义4.6),但是,由于表示时间复杂度的多项式太大了(即它们的次数太高了),在实际应用中它们几乎没有什么价值。这与有些具有非多项式时间复杂度(在4.6节定义)的算法在实用中形成了对比:这些非多项式时间复杂性的算法对有效求解困难问题的小规模实例仍然很有用(例如3.6.1节计算离散指数的Pollard袋鼠方法)。

我们将用**实际有效**这一术语来表示那些次数很小的多项式时间算法。例如,图灵机Div3、gcd算法、mod\_exp算法和Prime\_Test算法以及QKD协议都是实际有效的。现在看另外一个在现代密码学中广泛使用的实际有效的算法。

##### 4.4.6.1 有效算法:一个例子

概率素性检测的思想可以直接转化成一个算法,该算法能生成给定长度的随机**概率素数**。如果Prime\_Test( $n$ )返回YES,我们称 $n$ 是一个概率素数。算法4.7说明了如何生成这种给定长度的素数。

首先,假设Prime\_Gen( $k$ )会终止,这意味着该算法最终找到了一个数 $p$ ,满足Prime\_Test( $p$ )=YES(第2步)。根据我们对Prime\_Test的差错概率界估计,输出的 $p$ 不是素数的概率上界为 $2^{-k}$ ,其中 $k = \log_2 p$ 。

**算法 4.7** 随机  $k$  比特概率素数生成

输入 正整数;

(\* 输入写成输入的规模 \*)

输出 一个  $k$  比特随机素数。Prime\_Gen( $k$ )

1.  $p \in_U(2^{k-1}, 2^k - 1]$  且  $p$  为奇数;
2. 如果 Prime\_Test( $p$ ) = NO, return(Prime\_Gen( $k$ ));
3. return( $p$ )。

这里显然出现一个问题: Prime\_Gen( $k$ ) 最终能停下来吗?

有个著名的素数定理(例如, 参阅[172]的 28 页)表明, 小于  $X$  的素数大约有  $\frac{X}{\log X}$  个。因此  $k$  比特素数的个数大约为

$$\frac{2^k}{k} - \frac{2^{k-1}}{k-1} \approx \frac{2^k}{2k}$$

因此, 我们可以期望 Prime\_Gen( $k$ ) 会在第 2 步递归调用自身  $2k$  次才找到一个概率素数, 最终停了下来。

由于 Prime\_Test( $p$ ) 的时间复杂度以  $O_B((\log p)^4) = O_B(k^4)$  为界, 调用  $2k$  次 Prime\_Test 后, Prime\_Gen( $k$ ) 的时间复杂度界为  $O_B(k^5)$ 。

另外一个问题是: 尽管  $O_B(k^5)$  是关于  $k$  的一个多项式, 但它还是 Prime\_Gen( $k$ ) 算法输入长度的多项式吗? 即是否为关于  $k$  的长度的多项式?

对任意  $b > 1$ , 当我们将一个数  $n$  写成  $b$  进制表示时,  $n$  的长度是  $\log_b n$ , 这总是小于  $n$  的。为了使 Prime\_Gen( $k$ ) 为关于输入长度的多项式时间算法, 我们明确要求 Prime\_Gen( $k$ ) 的输入要写成输入的长度。使用一元编码或 1 进制表示, 实际上  $k$  写下来的长度还是  $k$ 。

**定义 4.7 数的一元表示** 正整数  $n$  的一元表示为

$$1^n = \underbrace{11 \cdots 1}_n$$

从现在开始, 我们将用 Prime\_Gen( $1^k$ ) 表示 Prime\_Gen 算法的一个调用实例。在本书的其余部分, 一个数的一元表示总是明确强调这个数的长度就是这个数本身。

## 4.5 非确定多项式时间

考虑下面的判定性问题:

**问题** SQUARE-FREENESS

输入  $N$ : 一个正的奇合数;

输出  $N$  无平方因子吗?

如果不存在素数  $p$  满足  $p^2 \mid N$ , 回答 YES。

SQUARE-FREENESS 问题很困难。到目前为止,还没有已知算法(不论是确定性算法还是概率算法)可以在关于输入长度的多项式时间内回答这一问题。当然,存在回答该问题的算法。例如下面就有一个:输入  $N$ ,用所有不超过  $\lfloor \sqrt{N} \rfloor$  的奇素数的平方穷举试除,如果所有试除都失败,则回答 YES。但是,对一般的输入实例  $N$ ,这种方法的运行时间为  $O(\lfloor \sqrt{N} \rfloor) = O(e^{\frac{\log N}{2}})$ ,即关于  $N$  的长度(的一半)的指数时间。

不过,不应认为 SQUARE-FREENESS 问题太难。如果我们知道问题的一些“内部信息”,称为证据(或证书或辅助输入),那么回答可以在关于输入长度的多项式时间内验证。例如,对输入  $N$ ,整数  $\phi(N)$  称为  $N$  的欧拉  $\phi$  函数,是小于  $N$  且与  $N$  互素的所有整数的个数(见 5.2.3 节的定义 5.11),它可以用做一个证据,使一个有效验证算法可以验证关于  $N$  是否无平方因子的回答。算法 4.8 就是一个有效的验证算法。

---

**算法 4.8** Square-Free( $N, \phi(N)$ )

1.  $d \leftarrow \gcd(N, \phi(N))$ ;
  2. 如果  $d = 1$  或  $d^2$  不整除  $N$ , 回答 YES; 否则回答 NO。
- 

熟悉  $\phi(N)$  含义的读者可以证明算法 4.8 的稳妥性(见习题 4.13)。这一验证算法归因于一个基本的数论事实,在第 6 章的 6.3 节就会很清楚该数论的结果。根据学习的最大公因子算法的时间复杂度(见 4.3.2.3 节),很显然上述算法运行时间为关于  $N$  的长度的多项式。

现在来描述一个计算设备:它给出求解和 SQUARE-FREENESS 问题有相同特性的一类问题的方法。该设备的计算可以用图 4.4 中的一棵树来描述。

该设备称为非确定性图灵机。它是图灵机的一个变型(回顾我们在 4.2 节对图灵机的描述)。在每一步,该机器有有限个选择作为其下一步。一个输入串称为可识别的条件是至少存在一系列合法移动,当机器扫描第一个输入符号时,它从机器初始状态开始,完成扫描输入串后到达满足终止条件的状态。这样的一系列移动称为一个识别序列。

可以这样想像,非确定性图灵机寻求可识别问题实例的解要进行一系列猜测:正确猜测的一系列移动形成一个识别序列。因此,机器可以做出的所有可能移动形成一棵树[称为一个非确定性图灵机的计算树,见图 4.4]。显然,树的大小(节点数)显然是关于输入规模的一个指数函数。但是,对于可识别的输入实例,由于一个识别序列中移动的次数就是树的深度  $d$ ,我们有  $d = O(\log(\text{树的节点个数}))$ ,因此识别序列中移动的次数必定以输入实例规模的多项式为界。所以,通过一系列正确猜测,识别可识别输入的时间复杂度是关于输入规模的一个多项式。

**定义 4.8 NP 类** NP 类表示用非确定性图灵机在多项式时间内可识别的语言类。

直接可以看出

$$P \subseteq NP$$

也就是说,  $P$  中的每一种语言(判定性问题)用非确定性图灵机是很容易识别的。也不难看出

$$ZPP, PP(\text{Monte Carlo}), PP(\text{Las Vegas}) \subseteq NP$$

事实上,  $ZPP$ 、 $PP(\text{Monte Carlo})$  和  $PP(\text{Las Vegas})$  都是真正的 NP 问题,因为它们实际上都是



**定义 4.9 复杂度上下界** 量  $B$  称为问题  $P$  的(复杂度)下界,如果求解  $P$  的任意算法  $A$  的复杂度开销都有  $C(A) \geq B$ 。

量  $U$  称为问题  $P$  的(复杂度)上界,如果求解  $P$  的任意算法  $A$  的复杂度开销都有  $C(A) \leq U$ 。

对  $P$  中的任何问题,通常可能(容易)确定它的复杂度下界,也就是说,给出一个精确多项式界表示求解该问题必须的步骤数。图灵机 Div3(例 4.1)给出了一个这样的例子:它识别一个  $n$  比特输入串恰好需要  $n$  步,即输入实例写的方法所允许的最少可能采用的步数。

但是,对于  $NP$  中的问题,总是难于确定复杂度下界,甚至找一个新的(即更小的)上界也很难。 $NP$  问题的所有已知复杂度界都是上界。例如,我们已经“证明” $\lfloor \sqrt{N} \rfloor$  是(通过试除)回答输入为  $N$  的 SQUARE-FREENESS 问题的一个复杂度上界。一个上界本质上是说:“求解这个问题只要这么多步就够了,”没明说的言外之意是:“但更少的步数是可能的。”事实上,对于 SQUARE-FREENESS 问题,式(4.6.1)给出的数域筛法分解  $N$  的复杂度需要的步数少于  $\lfloor \sqrt{N} \rfloor$ ,但它仍然是一个上界。

我们不应该混淆“下界”和“更小的界”。后者经常出现在文献中(例如, Cook 在他发现“可满足性问题”是“NP 完全”问题的著名论文[81]中就用过),表示新发现一个复杂度开销低于所有已知的开销(因此是一个更小的界)。即使确定一个更小的界(不是下界)通常也要求最小开销证明。确定出一个  $NP$  问题的下界意味着计算复杂性理论的一个重大突破。

对以复杂性理论为其安全性基础的现代密码学,确定  $NP$  问题的非多项式下界的困难性有重要影响。我们将在 4.8 节讨论这一点。

#### 4.5.1 非确定多项式时间完全

尽管不清楚是否有  $P = NP$ ,但我们确实知道  $NP$  中某些问题和  $NP$  中任何问题一样难。在这个意义上,如果我们有一个有效算法求解其中一个问题,那么求解  $NP$  中的任何问题都能够找到有效算法。这种问题称为非确定多项式时间完全的(NP 完全的或简记为 NPC)。

**定义 4.10 可多项式归约** 称语言  $L$  可多项式归约到另一种语言  $L_0$ ,如果存在一个确定性多项式时间界定的图灵机  $M$  可以将任何实例  $I \in L$  转化成另一个实例  $I_0 \in L_0$ ,满足  $I \in L$  当且仅当  $I_0 \in L_0$ 。

**定义 4.11 NP 完全** 语言  $L_0 \in NP$  是非确定性多项式时间完全的(NP 完全的),如果任意  $L \in NP$  都可以多项式归约到  $L_0$ 。

一个著名的 NP 完全问题是所谓的 SATISFIABILITY(可满足性)问题(由 Cook 发现[81]),它是所发现的第一个 NP 完全问题([229]的 344 页)。设  $E(x_1, x_2, \dots, x_n)$  表示由  $n$  个布尔变量  $x_1, x_2, \dots, x_n$  使用布尔运算符如  $\wedge$ 、 $\vee$  和  $\neg$  构成的一个布尔表达式。

**问题 SATISFIABILITY**

输入  $X = (x_1, \neg x_1, x_2, \neg x_2, \dots, x_n, \neg x_n);$   
 $E(x_1, x_2, \dots, x_n)。$

$E(x_1, x_2, \dots, x_n)$  的一个真值赋值是  $X$  的一个子列表  $X'$ , 满足对  $1 \leq i \leq n$ ,  $X'$  要么包含  $x_i$ , 要么包含  $\neg x_i$ , 但不能二者都包含, 且  $E(X') = \text{True}$ 。

问题  $E(x_1, x_2, \dots, x_n)$  可以满足吗?  
 也就是说, 存在一个真值赋值吗?  
 如果  $E(x_1, x_2, \dots, x_n)$  是可满足的, 回答 YES。

如果给定一个可满足的真值赋值, 那么显然 YES 回答可以在关于  $n$  的多项式时间内验证。因此, 根据定义 4.8, 我们知道 SATISFIABILITY  $\in \mathcal{NP}$ 。注意有  $2^n$  个可能的真值赋值, 到目前为止, 我们知道没有确定性多项式时间算法来确定它是否是可满足的。

证明 SATISFIABILITY 是 NP 完全的 (Cook [81]) 可以在 [9] 的第 10 章找到 (证明是构造性的, 它将任意一个非确定性多项式时间图灵机转化成一个求解 SATISFIABILITY 的图灵机)。

在 [120] 中列举了大量的 NP 完全问题。

对一个 NP 完全问题, 新找到的任何更小的上界可以多项式“归约”(转化) 为整个 NP 问题类的一个新结果。因此, 如 [99] 建议, 非常希望设计安全性基于 NP 完全问题的密码算法。对这种密码体制的一个成功攻击有望导致解决整类困难问题, 这应该是不可能的。然而, 不管是实现一个安全实用的密码算法还是用对这种密码体制的一个攻击来求解整类 NP 完全问题, 这样一个合理的想法到目前为止还没有富有成效的结果。我们将在 4.8.2 节中讨论这一看起来有点奇怪的现象。

## 4.6 非多项式界

有许多函数大于任意多项式。

**定义 4.12 非多项式有界量** 函数  $f(n): \mathbb{N} \rightarrow \mathbb{R}$  称为是关于  $n$  的任意多项式无界的, 条件是对任意多项式  $p(n)$ , 存在一个自然数  $n_0$ , 使得对任意  $n > n_0$ , 都有  $f(n) > p(n)$ 。

函数  $f(n)$  称为多项式有界的, 条件是它不是非多项式有界量。

**例 4.3** 证明对任意  $a > 1, 0 < \epsilon < 1$ , 函数

$$f_1(n) = a^{n^{(\log n)^{1-\epsilon}}}, \quad f_2(n) = n^{(\log \log \log n)^\epsilon}$$

关于  $n$  的任意多项式无界。

设  $p(n)$  为任意多项式,  $d$  表示它的次数,  $c$  表示它的最大系数, 则  $p(n) \leq cn^d$ 。首先, 令  $n_0 = \max\left(c, \left\lfloor \left(\frac{d+1}{\log a}\right)^{\frac{2}{\epsilon}} \right\rfloor\right)$ , 那么对任意  $n > n_0$ , 有  $f_1(n) > p(n)$ 。其次, 令  $n_0 = \max\left(c, \left\lfloor \exp(\exp(\exp((d+1)^{\frac{1}{\epsilon}}))) \right\rfloor\right)$ , 则对任意  $n > n_0$ , 有  $f_2(n) > p(n)$ 。□

与(确定性或非确定性)多项式时间问题相比, 时间复杂度为非多项式时间界的问题视为在计算上是难处理的或不可行的。这是因为, 当问题实例的规模增长时, 求解这种问题所要求的资源增长得太快, 以至于很快就大得不实际了。例如, 设  $N$  是一个长度为  $n$  的合数 (即  $n = \log N$ ); 那么例 4.3 中取  $a \approx \exp(1.922\,999\,4 \cdots + o(1))$  (其中  $o(1) \approx \frac{1}{\log N}$ ) 和  $\epsilon = \frac{1}{3}$ , 函数  $f_1(\log N)$  用数域筛法分解  $N$  的时间复杂度表达式为 (例如, 见 [71]):

$$\exp(1.922\,999\,4 \cdots + o(1)) (\log N)^{\frac{1}{3}} (\log \log N)^{\frac{2}{3}} \quad (4.6.1)$$



式(4.6.1)是关于  $N$  的亚指数表示(sub-exponential expression)。如果  $\frac{1}{3}$  替换为 1, 那么该式成为指数表示。亚指数函数比指数函数增长缓慢得多, 但比多项式函数快得多。对 1024 比特的整数  $N$ , 式(4.6.1)是一个大于  $2^{86}$  的量, 即使使用大量的计算机并行运行, 这样大的量也是无法处理的。亚指数时间复杂度也适用于数量级  $N$  的有限域中解“离散对数问题”的最好算法(见 8.4 节的定义 8.2)。

但是, 我们应该注意到用于定义 4.12 中函数比较的渐近(asymptotic)模式(在定义 4.12 中,  $f(n)$  也称为渐近大于任意多项式, 或对于充分大的  $n$ , 大于关于  $n$  的任意多项式), 尽管  $f(n)$  不以关于  $n$  的任何多项式为界, 但通常情况下, 对于很大的数  $n_0$  和  $n \leq n_0$ ,  $f(n)$  还是可能小于某个多项式  $p(n)$ 。例如, 对于例 4.3, 尽管对任何  $d \geq 1$ ,  $f_2(n)$  渐近大于  $n^d$ , 而如果取  $\epsilon = 0.5$ , 对所有的  $n \leq 2^{742762245454927736743541}$ , 函数  $f_2(n)$  都是小于平方函数  $n^2$  的量。这就是实际中为什么一些非多项式界时间复杂度的算法对求解输入规模小的问题仍然很有效的原因。在 3.6.1 节中已经介绍的 Pollard  $\lambda$  方法求解小的离散对数就是这样的算法。

在使用阶号(见 4.3.2.4 节的定义 4.2)时, 我们故意忽略了复杂度表示中的所有常系数, 但是应该注意常系数出现在非多项式界量指数位置上的影响[例如式(4.6.1)中的  $1.922\ 999\ 4 \cdots + o(1)$ ]。例如, 如果一个新的整数分解算法改进了 NFS 算法, 将表达式中的常指数  $1.922\ 999\ 4$  减小到了 1, 那么使用该算法分解 1024 比特合数的时间复杂度将从大约  $2^{86}$  降到大约  $2^{45}$ 。后者对今天的计算技术来说不再是太大的量。尤其是对于 NFS 方法, 当前加速该方法的一个研究努力就是缩小常数指数, 例如通过时间空间折中(尽管减小时间开销可能要增加存储开销, 确实还是有可能将时间复杂度减小到一定程度)。

对很大的量, 我们已经定义了非多项式界的概念; 对很小的量, 我们也可以定义一个概念。

**定义 4.13 可忽略量** 函数  $\epsilon(n): \mathbb{N} \rightarrow \mathbb{R}$  称为关于  $n$  的一个可忽略量(或称  $\epsilon(n)$  是可忽略的), 条件是其倒数  $\frac{1}{\epsilon(n)}$  是关于  $n$  的一个非多项式有界的量。

例如, 对任意多项式  $p$ ,  $\frac{p(n)}{2^n}$  就是一个可忽略量。由于这个原因, 我们有时候说  $p(n)$  点集在集合  $\{1, 2, 3, \dots, 2^n\}$  中(关于后者)只有可忽略比例的数目, 或者说  $\{1, 2, 3, \dots, 2^n\}$  中的  $p(n)$  个点在该集合中是稀疏的。

如果  $\epsilon$  是一个可忽略量, 那么  $1 - \epsilon$  就称为一个压倒性的量。因此, 例如, 我们也说  $\{1, 2, \dots, 2^n\}$  的任意非稀疏(即稠密)子集包含点的个数占有压倒性比例。

可忽略量比任意多项式的倒数更快地趋于 0。如果我们认为非多项式有界量是不可处理的(例如在资源分配方面), 那么忽略任何非多项式有界量的倒数级的量应该是无关紧要的。

更多的例子如:

$$\text{Prob}[\text{"Prime\_Gen}(1^k) \text{ 不是素数"}]$$

关于  $k$  是可忽略的, 且

$$1 - \text{Prob}[\text{"Prime\_Gen}(1^k) \text{ 不是素数"}] = \text{Prob}[\text{"Prime\_Gen}(1^k) \text{ 是素数"}]$$

关于  $k$  是压倒性的。

回顾例 3.6, 对  $k$  比特素数  $p$  ( $q = \frac{p-1}{2}$  也是一个素数), 我们可以忽略  $\frac{1}{p-1}$  数量级的量, 因此得到  $\text{Prob}[A] \approx \frac{3}{4}$ 。

最后,如果一个量不是可忽略的,那么常常称它为一个不可忽略量或显著量。例如通过一系列例子我们已经知道,对于成员关系可以有效判定的 $PP$ 类中的判定性问题,通过对计算树(见图4.4)空间随机取样,找出一个证据证实成员关系就有显著概率。

## 4.7 多项式时间不可区分性

我们刚才认为忽略一个可忽略的量是无关紧要的。但是有时候当忽略一个量时,我们会感到无可奈何,因为我们不得不放弃不忽略该量的想法。现在通过一个例子来描述这一情形。

考虑在固定长度的大奇合数空间中进行两个试验,一个称为  $E_{2\_Prime}$ , 另一个称为  $E_{3\_Prime}$ 。这两个试验产生同样长度的随机大整数:  $E_{2\_Prime}$  产生的每一个大整数都是两个不同素因子的乘积;  $E_{3\_Prime}$  产生的每一个大整数都是三个或更多个不同素因子的乘积。现在,假设有人根据其中一个试验给你一个整数  $N$ 。你有把握判断  $N$  是由哪一个试验中产生的吗?(回忆一下,  $E_{2\_Prime}$  和  $E_{3\_Prime}$  产生的整数同样长)。

根据定义3.5(见3.5节),这种试验结果是试验内部随机移动的一个随机变量。我们知道  $E_{2\_Prime}$  产生的随机变量和  $E_{3\_Prime}$  产生的随机变量的概率分布很不相同:  $E_{2\_Prime}$  以概率1产生两个素数的乘积,而对  $E_{3\_Prime}$  却永远都不会发生这种情况。但是,要区分由这两个试验产生的随机变量事实上是非常困难的。

现在我们来精确定义不可区分总体(也称为不可区分试验)的含义。

**定义4.14 总体分辨器** 设  $E = \{e_1, e_2, \dots\}$ ,  $E' = \{e'_1, e'_2, \dots\}$  是两个总体,其中,  $e_i, e'_j$  是有限样本空间  $S$  中的随机变量。记  $k = \log_2 \#S$ , 设  $a = (a_1, a_2, \dots, a_\ell)$  是由  $E$  或者由  $E'$  生成的随机变量,  $\ell$  关于  $k$  的多项式有界。

$(E, E')$  的分辨器  $D$  是一个概率算法, 在关于  $k$  的多项式时间内终止, 其输出属于  $\{0, 1\}$ , 满足 (i)  $D(a, E) = 1$  当且仅当  $a$  由  $E$  产生; (ii)  $D(a, E') = 1$  当且仅当  $a$  由  $E'$  产生。

称  $D$  以优势  $\text{Adv} > 0$  区分  $(E, E')$ , 如果

$$\text{Adv}(D) = |\text{Prob}[D(a, E) = 1] - \text{Prob}[D(a, E') = 1]|$$

注意描述分辨器  $D$  的优势时使用概率分布是很重要的: 分辨器是一个概率算法, 并且还是一个多项式时间算法; 它的输入具有多项式有界的规模。

许多随机变量可以很容易区分, 这里就有一个例子。

**例4.4** 设  $E = \{k \text{ 比特素数}\}$  和  $E' = \{k \text{ 比特合数}\}$ 。定义  $D(a, E) = 1$  当且仅当  $\text{Prime\_Test}(a) \rightarrow \text{YES}$ ;  $D(a, E') = 1$  当且仅当  $\text{Prime\_Test}(a) \rightarrow \text{NO}$  ( $\text{Prime\_Test}(a)$  详细说明在算法4.5中)。那么  $D$  是  $E$  和  $E'$  的分辨器。当  $a \in E$ , 我们有  $\text{Prob}[D(a, E) = 1] = 1$  且  $\text{Prob}[D(a, E') = 1] = 0$ ; 当  $a \in E'$  时, 有  $\text{Prob}[D(a, E) = 1] = 2^{-k}$  且  $\text{Prob}[D(a, E') = 1] = 1 - 2^{-k}$ 。因此,  $\text{Adv}(D) \geq 1 - 2^{-(k-1)}$ 。□

**定义4.15 多项式时间不可区分性** 设总体  $E, E'$  和安全参数  $k$  如定义4.14。如果对所有充分大的  $k$  及关于  $k$  不可忽略的优势  $\text{Adv} > 0$ , 不存在  $(E, E')$  的分辨器, 则  $E, E'$  称为是多项式不可区分的。

在计算复杂性理论中, 广泛认为下述假设似乎是合理的。

**假设 4.1 一般不可区分性假设** 存在多项式不可区分总体。

总体  $E_{2\_Prime}$  和  $E_{3\_Prime}$  被认为是多项式不可区分的。换句话说,如果有人给我们多项式个整数,它们或者都是由  $E_{2\_Prime}$  产生,或者都是由  $E_{3\_Prime}$  产生,即使用最好的算法作为分辨器,我们也会很快就会感到无望,不得不放弃分辨的尝试。

注意,如果我们能够分解  $N$ ,那么就能正确地回答该问题,优势  $Adv$  必定不小于式(4.6.1)中函数的倒数。但是这个值太小了,以至于只好忽略掉。我们说我们不可能分辨两个总体,因为即使用最好的分辨器,其优势关于这两个总体产生的整数长度也是可忽略的。这样的优势关于我们的计算资源是一个缓慢增加的函数。这里“缓慢增加”的意思是,即使极大地增加计算资源,优势也只增加微小的一点,以至于很快就感到无望了。

对许多密码学算法和协议,多项式不可区分性是一个重要的安全准则。在现代密码学中,有许多实用方法构造有用的多项式不可区分总体。例如,伪随机数生成器是密码学的一个重要组成部分,这种生成器生成伪随机数,它的分布完全由一个种子确定(即以确定的方式)。然而,好的伪随机数生成器生成的随机数和真随机数是多项式不可区分的,也就是说,从这种生成器输出的随机变量和长度相同的均匀随机分布串二者的分布是不可区分的。事实上,下面的假设是假设 4.1 的一个实例。

**假设 4.2 (伪随机性和真随机性之间的不可区分性)** 存在伪随机函数和真随机函数是多项式不可区分的。

在第 8 章将看到一个伪随机函数(伪随机数生成器),它和均匀分布是多项式不可区分的。在第 14 章将进一步学习一种著名的公钥密码体制,即 Goldwasser-Micali 密码体制;该密码体制的安全性基于与  $E_{2\_Prime}$  和  $E_{3\_Prime}$  相关的多项式不可区分总体(在 6.5.1 节将讨论它们的关系)。另一个例子是,有些阿贝尔群中四个元素组成的一个 Diffie-Hellman 元组(13.3.4.3 节的定义 13.1)和同一个群中的随机四元组构成不可区分总体,为 ElGamal 密码体制和许多零知识证明协议提供了安全基础。后面几章我们将经常用到多项式不可区分概念。

## 4.8 计算复杂性理论与现代密码学

在计算复杂性这一简短课程结束时,我们将讨论计算复杂性和现代密码学的关系。

### 4.8.1 必要条件

一方面可以说,基于复杂性理论的现代密码学将  $P \neq NP$  作为一个必要条件,我们称之为  $P \neq NP$  猜想<sup>①</sup>。

另一方面,加密算法应向拥有正确加/解密密钥的用户提供有效的加/解密算法,而给试图从密文中提取明文或不用正确密钥构造合法密文的人(攻击者或密码分析者)制造一个难处理问题。因此,密码学密钥对基于 NP 问题的密码体制来说起着—个证据或辅助输入(更准确的称呼)的作用。

也许有人想驳斥我们关于基于复杂性理论的密码学必要条件的论断,认为也许存在一种密码体制基于  $P$  中的非对称问题:加密是一个  $O(n)$  算法且最好的破译算法的阶也为  $O(n^{100})$ 。事

<sup>①</sup> 最近调查显示,绝大多数计算机理论科学家相信  $P \neq NP$ 。

实上,即使对很小的  $n = 10$  的情形,  $O(n^{100})$  也是  $2^{232}$  数量级的,远远超出了全世界最先进计算技术组合起来的能力。因此,如果这种多项式时间密码体制存在,即使证明了  $P = NP$ ,我们所处的形势也不坏。但是问题在于,尽管对任意的  $k$ ,  $P$  确实包含  $O(n^k)$  复杂度的问题,它却不包含具有非对称复杂度行为的问题。对  $P$  中任意给定的问题,如果一个长度为  $n$  的实例可以在时间  $n^k$  内求解,那么对任意  $\alpha > 0$ ,由于算法的确定性行为,时间  $n^{k+\alpha}$  都是不必要的。

该猜想还构成了存在单向函数的必要条件。在本书的开始(1.1.1节),我们假设单向函数具有“神奇特性”(性质 1.1):对所有整数  $x$ ,从  $x$  计算  $f(x)$  是容易的,但是,除了问题中可忽略的部分实例,给定绝大多数的  $f(x)$  值,要找出  $x$  是极为困难的。现在我们知道  $NP$  类提供了候选问题来实现具有“神奇特性”的单向函数。例如,可满足性问题就定义了一个从  $n$  元布尔空间到  $\{\text{True}, \text{False}\}$  的单向函数。

进一步,存在单向函数构成存在数字签名的必要条件。数字签名应具有这样的性质:易于验证却难于伪造。

而且,我们在 4.7 节学习的多项式时间不可区分性概念也是基于  $P \neq NP$  的猜想。这是  $NP$  中困难问题的判定性情形。在第 14 章、第 15 章和第 17 章将看到多项式时间不可区分性在现代密码学——密码学算法和协议的正确性中的重要作用。

我们应该特别提及  $P \neq NP$  猜想在公钥密码学中一个令人着迷的课题:零知识证明协议 [128] 和交互式证明系统中根本性的重要作用。

零知识证明协议是一种交互式程序,在两个主体之间运行,分别称为示证者和验证者,后者只有多项式界的计算能力。因为前者拥有辅助输入,协议允许前者向后者证明前者知道一个 NP 问题的 YES 答案(例如,问题 SQUARE-FREENESS 或“ $N$  是  $E_{2\text{-Prime}}$  生成的吗?”的 YES 答案),而不让后者知道怎样进行这样的证明(即不向后者泄露辅助输入)。因此验证者关于示证者的辅助输入获得的是“零知识”。这样一个证明可以用非确定性图灵机增加一条随机纸带来模拟。示证者可以利用辅助输入,因此(示证者)可以指令图灵机关于输入问题沿着识别序列移动(证明 YES 答案)。所以,证明的复杂度是关于输入实例规模的一个多项式。验证者应当向示证者提问,要求示证者指令图灵机或者沿着识别序列或者沿着另外一个不同序列移动,而且提问应该是均匀随机的。因此在验证者看来,该证明系统正好按随机化图灵机(回顾 4.4 节)的模式运行。这种随机化图灵机的差错概率通过独立重复执行可以减小到一个可忽略量,事实上,正是这一特性构成了验证者相信示证者确实知道输入问题 YES 答案的基础。

$P \neq NP$  猜想在零知识协议中起到下面两个作用:(i) NP 问题的辅助输入使得示证者可以进行有效证明,(ii) 问题的困难性意味着验证者自己不能验证示证者的声明。我们将在第 18 章学习零知识证明协议。

#### 4.8.2 非充分条件

另一方面,即使密码体制基于 NP 完全问题,  $P \neq NP$  猜想也并没有提供这种密码体制安全性的充分条件。NP 完全的著名背包问题破解就给出了一个反例 [202]。

在我们学习关于计算复杂性的课程之后,对于为什么基于 NP(或者甚至 NP 完全)问题的密码体制经常被攻破,我们现在可以给出两个简要而清晰的解释。

首先,正如在这门课程开始阶段(例如,回顾定义 4.1)指出的,计算复杂性的复杂性理论方法用了全称量词:“任意实例  $I \in L$ ”来限制复杂性类中的语言  $L$ (问题),这就导致了最坏情

**况复杂性分析:**即使一个问题只有可以忽略的少数困难实例,该问题也被认为是困难的。相反,密码分析只要能破解不可忽略比例的实例,就认为是成功的。这就是为什么破解一个基于 NP 完全问题的密码体制未必导致其基于的 NP 完全问题的求解。显然,对衡量实用密码体制的安全性,最坏情况的复杂性准则是做不到的,也没有用处。

第二个解释是确定 NP 问题一个新的更小上界的内在困难性(注意,术语“新的更小上界”对 NP 问题是有意义的,回顾 4.5 节中对上下界的讨论)。对于基于 NP 问题的密码体制的安全性基础,即使已经证明该基础是它基于的 NP 问题的难处理性,最好的情形也不过是基于一个公开问题,因为我们只知道该问题的一个复杂性上界。而且更常见的情况是,对于这样一种基于 NP 问题的密码体制,甚至连其基于的难处理性也没有明确地确定。

将复杂性理论中的困难性作为现代密码学的安全性基础是不充分的,对该不充分性的进一步研究是本书的主题:应用密码学非教科书式安全性方面的问题(回顾 1.1.3 节)。可能有许多实际的方式危及到实际应用的密码系统,它们可能和构成算法安全性基础的数学上的难处理性没有什么关系。本书的以后章节会给出大量的解释和证据表明这一点。

对于安全密码体制的设计和分析,有一种积极态度近年来逐渐得到广泛接受,这就是使用多项式归约技术(见定义 4.10)形式化证明一种密码体制的安全性(可证明安全性:通过有效的转化,将对密码体制的任何有效攻击归约到解一类已知 NP 问题的一个实例。通常,该类 NP 问题是广泛接受的“谱系类”中的一小部分。这种归约通常称为归约为矛盾,因为人们广泛相信,广为接受的“谱系问题”没有有效解法。对于考虑中的密码体制的安全性,这样的证明提供了很高的可信度。我们将在第 14 章和第 15 章学习这一方法。

## 4.9 本章小结

计算复杂性是现代密码学的一个基础(事实上是最重要的基础)。由于其重要性,本章独立系统地介绍了这一基础。

作为可计算的问题类,我们首先介绍了图灵可计算性的概念,该类中有的问题是易处理的(在多项式时间内有效求解),它们或者是确定性的(属于  $P$ ),或者是非确定性的(概率多项式时间类  $PP$  的几个子类),其他的问题是难处理的( $NP$  类,仍然是  $PP$  中的子类,这在 18.2.3 节就会很清楚)。不论是确定的还是别的算法, $NP$  中的问题似乎不可能用有效的算法求解,而给定一个证据,它们关于类的成员资格可以有效地验证。

我们在课程中还介绍了计算复杂性及其在现代密码学应用中的重要概念,包括有效算法(构造了几个重要的算法,并进行了精确的时间复杂度分析)、阶号、多项式可归约性、可忽略量、上界、下界和非多项式界、不可区分性。这些概念在本书以后章节会经常用到。

最后,我们讨论了  $NP$  问题和复杂性理论基础在现代密码学中的重要作用。

## 习题

- 4.1 构造一个识别偶数的图灵机,然后构造一个图灵机识别被 6 整除的整数。  
提示:第二个图灵机可以用一个运算表合取第一个图灵机的运算表和图 4.2 中 Div3 的运算表。
- 4.2 在度量一个算法的计算复杂度时,为什么比特复杂度,即计算比特运算次数,比计算例如整数乘法次数的度量更可取?



提示:考虑一个问题可能有不同规模的实例。

- 4.3 定理 4.1 给出  $\gcd(x, y)$  (对  $x > y$ ) 的开销为  $\log x$  次模运算, 由于模运算和除法的开销相同, 都是  $O_B((\log x)^2)$ , 这样得出  $\gcd(x, y)$  的开销为  $O_B((\log x)^3)$ 。但是, 在标准的教科书中  $\gcd(x, y)$  的开销为  $O_B((\log x)^2)$ 。我们的度量有什么不同?

提示:观察不等式(4.3.12)。

- 4.4 证明定理 4.2 中的第 2 条和第 3 条性质。

- 4.5 证明  $\mathcal{PP}(\text{Monte Carlo})$  和  $\mathcal{PP}(\text{Las Vegas})$  是互补的 [表示成  $\mathcal{PP}(\text{Monte Carlo}) = \text{co } \mathcal{PP}(\text{Las Vegas})$ ]。也就是说, 识别  $I \in L$  的 Monte Carlo 算法就是识别  $I \in \bar{L}$  的 Las Vegas 算法, 反之亦然。用同样的方法证明  $\mathcal{BPP} = \text{co } \mathcal{BPP}$ 。

- 4.6 在计算复杂性文献中, 我们经常看到  $\mathcal{BPP}$  类用  $\epsilon = \frac{2}{3}$  [对于式(4.4.1)] 和  $\beta = \frac{1}{3}$  [对于式(4.4.1)] 定义。我们用的是任意常数  $\epsilon \in [\frac{1}{2} + \alpha, 1)$ ,  $\delta \in (0, \frac{1}{2} - \beta]$ , 其中  $\alpha > 0, \beta > 0$ 。这两种不同的定义方式有区别吗?

- 4.7 证明: 在式(4.4.5)中, 对于  $\epsilon(k)$ , 当  $k \rightarrow \infty$  时,  $\epsilon(k) \rightarrow 1$ 。

提示:  $1 - \epsilon(k) \rightarrow 0$ 。

- 4.8 解释在  $\mathcal{BPP}$  的差错概率特征中, 为什么差错概率必须明显偏离  $\frac{1}{2}$ , 即 (4.4.11) 中的  $\alpha$  和  $\beta$  为什么必须是非零常数?

提示: 考虑一枚“有偏”硬币: 一面比另一面以一个可忽略的量更可能出现。通过抛掷硬币和使用大数判别准则, 你能找出更可能出现的那一面吗?

- 4.9 在我们度量 QKD 协议(协议 4.1)的稳妥性差错概率时, 提到了 Eve 的两个策略: 向 Bob 发送  $m$  个全新的光子状态, 或转发她观察到的任何状态。我们只计算了 Eve 采用后一个策略时的稳妥性差错概率。请用前一个策略导出同样的稳妥性差错概率结果。

- 4.10 对正整数  $n$ , 我们用  $|n| = \log_2 n$  表示  $n$  的长度(它是  $n$  的二进制表示的比特数)。但是在大多数情况下,  $n$  的长度写做  $\log n$ , 没有明确说明它的底数(省略的底数是自然底数  $e$ )。证明: 对任意底数  $b > 1$ ,  $\log_b n$  是  $n$  长度的正确度量, 即对任意底数  $b > 1$ , 陈述“关于  $n$  长度的多项式”保持不变。

- 4.11 除了上面提到的问题以外, 我们有时候将一个正数表示成一元编码形式, 即用  $1^n$  表示  $n$ , 这为什么是必要的?

- 4.12 什么是有效算法? 什么是实际有效算法?

- 4.13 如果你已熟悉欧拉  $\phi$  函数(将在 6.3 节中介绍)的性质, 请证明算法 4.8 的正确性。

- 4.14 给出两个不可区分总体的例子。

- 4.15 为什么安全性基于 NP 完全问题的密码体制不一定是安全的?

- 4.16 说出下列问题的区别与联系:

- i) 图灵可计算的
- ii) 难处理的
- iii) 易处理的
- iv) 确定性多项式时间
- v) 实际有效的



## 第5章 代数学基础

### 5.1 引言

密码学算法和协议中,消息是作为有限空间中的数字或元素来处理的。编码(加密)和所需解码(解密)的各种操作必须在消息之间进行变换,以使变换服从有限消息空间内部的封闭性。然而,数的一般运算诸如我们所熟悉的加、减、乘、除并不满足有限空间(整数或是一个区间里的数)内部的封闭性。因此,仅仅使用所熟悉的数的一般运算通常并不能构造在有限消息空间里运行的密码算法。相反,密码算法通常运行于具有某些保持封闭性的代数结构的空间中。

本章介绍三种代数结构,这三种代数结构不仅是抽象代数的中心概念,而且为现代密码学和密码协议提供了基本元素和运算。这三种结构就是群、环和域。

#### 5.1.1 章节纲要

我们在5.2节中讨论群,在5.3节中讨论环和域,在5.4节讨论有限域的结构。最后在5.5节中我们用椭圆曲线上的点来构造有限群。

### 5.2 群

粗略来讲,群是一个对象集合,在这个集合中任意两个对象之间定义了一种运算。在对象集合上定义运算是很自然的,例如在古代,每天日落时分,牧羊人将清点他的羊群,也许这位牧羊人根本不识数,但这并不妨碍他正确地完成任务。他会随身带上一袋鹅卵石,将每一只羊与每一个鹅卵石相匹配,那么只要每次结束匹配的时候没有剩余的鹅卵石,他就知道他的羊群完好无损。这样,牧羊人实际上构造了一个群,在这个群上定义的运算就是“加1”运算。在这里,无论是羊、鹅卵石或是一些其他对象,重要的是在一个对象集中进行了一种运算,并且得到一个结果,而这个结果仍然在这个集合里。

**定义 5.1 群** 集合  $G$  和运算  $\circ$  一起称为群  $(G, \circ)$ , 前提是运算满足下面的条件:

1.  $\forall a, b \in G$ , 有  $a \circ b \in G$  (封闭律)
2.  $\forall a, b, c \in G$ , 有  $a \circ (b \circ c) = (a \circ b) \circ c$  (结合律)
3. 存在唯一的元素  $e \in G$ , 使得  $\forall a \in G$ , 均有  $a \circ e = e \circ a = a$  (单位元律)  
元素  $e$  称为单位元。
4.  $\forall a \in G$ , 存在  $\exists a^{-1} \in G$ , 使得  $a \circ a^{-1} = a^{-1} \circ a = e$  (可逆律)

在表示群  $(G, \circ)$  时,我们通常省略运算符号  $\circ$ , 而用  $G$  来表示一个群。

**定义 5.2 有限群和无限群** 如果集合  $G$  中的元素个数是有限的,那么群  $G$  就称为有限群,否则称为无限群。

**定义 5.3 阿贝尔群** 如果对所有的  $a, b \in G$ , 均有  $a \circ b = b \circ a$ , 则群  $G$  称为阿贝尔群。

换句话说, 阿贝尔群就是交换群(commutative group)。由于本书不讨论非阿贝尔群, 所以本书其余部分出现的所有群均指阿贝尔群, 我们通常省略前缀“阿贝尔”。

### 例 5.1 群

1. 整数集  $\mathbb{Z}$  在加法  $+$  下构成群, 即  $(\mathbb{Z}, +)$  是一个群, 其中  $e = 0, a^{-1} = -a$ 。这是一个加法群, 并且是一个无限群(且是阿贝尔群)。同样, 有理数集  $\mathbb{Q}$ 、实数集  $\mathbb{R}$  和复数集  $\mathbb{C}$  都是无限加法群, 单位元和逆元的定义同上。
2.  $\mathbb{Q}$ 、 $\mathbb{R}$  和  $\mathbb{C}$  中的非零元素在乘法下构成群, 其中  $e = 1, a^{-1}$  就是乘法逆元(通常方式下定义的)。我们将这些群分别记为  $\mathbb{Q}^*$ 、 $\mathbb{R}^*$  和  $\mathbb{C}^*$ 。因此, 这些群的完整表示就是:  $(\mathbb{Q}^*, \cdot)$ 、 $(\mathbb{R}^*, \cdot)$  和  $(\mathbb{C}^*, \cdot)$ 。这些群称为乘法群, 都是无限群。
3. 对任意  $n \geq 1$ , 整数模  $n$  集合构成一个包含  $n$  个元素的有限加法群, 这里的加法指模  $n$  加法, 单位元是 0, 对群中任一元素  $a, a^{-1} = n - a$  (4.3.2.5 节中定理 4.2 的性质 2), 我们将这个群记为  $\mathbb{Z}_n$ 。因此, 这个群的完整表示是  $(\mathbb{Z}_n, +(\bmod n))$ 。(注意,  $\mathbb{Z}_n$  是正式和标准记法  $\mathbb{Z}/n\mathbb{Z}$  的简化表示, 我们将在例 5.5 中给出理由。)
4. 时钟上表示小时的数字在模 12 加法下构成  $\mathbb{Z}_{12}$ , 我们把  $(\mathbb{Z}_{12}, +(\bmod 12))$  称为“时钟群”。
5.  $\mathbb{Z}_n$  中包含所有与  $n$  互素的元素(即  $\gcd(a, n) = 1$ )的子集构成一个有限乘法群, 这里乘法指模  $n$  乘法,  $e = 1$ , 对群中任一元素  $a, a^{-1}$  可以用扩展欧几里得算法(算法 4.2)计算。我们用  $\mathbb{Z}_n^*$  表示这个群。例如,  $(\mathbb{Z}_{15}^*, (\bmod 15)) = (\{1, 2, 4, 7, 8, 11, 13, 14\}, \cdot(\bmod 15))$ 。
6. 集合  $B = \{F, T\}$ , 令  $\circ = \oplus$  (逻辑异或)是:  $F \oplus F = F, F \oplus T = T \oplus F = T, T \oplus T = F$ 。则  $B$  在  $\oplus$  下是一个有限群, 其中  $e = F, T^{-1} = T$ 。
7.  $x^3 - 1 = 0$  的根在乘法运算下构成一个有限群, 其中  $e = 1$  (1 显然是一个根), 用  $\text{Root}(x^3 - 1)$  表示此群。让我们来找出群  $\text{Root}(x^3 - 1)$  中的其他元素和它们的逆。由于三次多项式  $x^3 - 1$  只有三个根, 令  $\alpha, \beta$  表示另两个根。由于  $x^3 - 1 = (x - 1)(x^2 + x + 1)$ , 则  $\alpha, \beta$  必是  $x^2 + x + 1 = 0$  的两个根。由二次方程根与系数的关系, 我们有  $\alpha\beta = 1$ 。因此,  $\alpha^{-1} = \beta$  且  $\beta^{-1} = \alpha$ 。读者可以验证运算满足封闭律(即  $\alpha^2$  和  $\beta^2$  均为  $x^3 - 1 = 0$  的根)。

□

**定义 5.4 重复群运算的简化表示** 令  $G$  是运算  $\circ$  下的一个群, 对任一元素  $a \in G$  和任一非负整数  $i \in \mathbb{N}$ , 我们将下面的元素

$$\underbrace{a \circ a \circ \cdots \circ a}_i$$

记为  $a^i \in G$ 。

我们应该注意以下两点注释。

#### 注释 5.1

- i)  $a^i \in G$  只是  $\underbrace{a \circ a \circ \cdots \circ a}_i$  的一个简化表示, 注意整数  $i$  和元素  $a$  之间的“运算”并不是群运算。

ii) 一些群习惯上写成加法群,例如 $(\mathbb{Z}_n, +(\bmod n))$ 。对于这些群,读者可以把 $a^i$ 看做是 $i \cdot a$ 。然而必须注意到,简化写法中的“ $\cdot$ ”并不是一个群运算,整数 $i$ 通常也不是一个群元素[考虑 $(\mathbb{Z}_n, +(\bmod n))$ 的例子,其中 $i > n$ ]。□

**定义 5.5 子群** 如果群 $G$ 的非空子集 $H$ 在与 $G$ 同样的运算下自身构成一个群,我们就把 $H$ 称为群 $G$ 的一个子群。用 $H \subseteq G$ 来表示 $H$ 是 $G$ 的一个子群,而 $H \subset G$ 则表示 $H$ 是 $G$ 的一个真子群(即 $H \neq G$ )。

### 例 5.2

1. 在加法运算下, $\mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$ 。
2. 在加法运算下,所有偶数与0构成的集合是(1)中所有群的一个子群<sup>①</sup>。
3. “时钟群” $(\mathbb{Z}_{12}, +(\bmod 12))$ 有下面的子群: $(\{0\}, +)$ ,  $(\{0, 6\}, +)$ ,  $(\{0, 4, 8\}, +)$ ,  $(\{0, 3, 6, 9\}, +)$ ,  $(\{0, 2, 4, 6, 8, 10\}, +)$ ,  $(\mathbb{Z}_{12}, +)$ 。
4. 在乘法运算下, $\mathbb{Q}^* \subseteq \mathbb{R}^* \subseteq \mathbb{C}^*$ 。
5. 令 $n$ 是一个正奇数, $\text{Fermat}(n)$ 表示 $\mathbb{Z}_n^*$ 的一个子集,使得对于 $\text{Fermat}(n)$ 中任一元素 $a$ ,均满足 $a^{\frac{n-1}{2}} \equiv \pm 1 (\bmod n)$ ,那么 $\text{Fermat}(n) \subseteq \mathbb{Z}_n^*$ 。并且如果 $n$ 是一个素数,那么由费马小定理(6.4节中的定理6.10), $\text{Fermat}(n) = \mathbb{Z}_n^*$ ;否则 $\text{Fermat}(n)$ 是 $\mathbb{Z}_n^*$ 的一个真子群。
6.  $\{F\}$ 是例5.1(6)中群 $B$ 的一个真子群,由于 $\{T\}$ 不包含单位元,所以 $\{T\}$ 不是 $B$ 的子群(也就是不满足单位元律)。
7. (见例4.1)多项式时间语言 DIV3 是 $\mathbb{Z}$ 的一个子群。
8. 集合 $\{e\}$ 是任意群的一个子群。□

**定义 5.6 群的阶** 有限群 $G$ 中元素的个数称为 $G$ 的阶,记为 $\#G$ 。

### 例 5.3

1.  $\#Z_n = n$ ;
2. 在例5.1(6)中, $\#B = 2$ ;
3. 在例5.1(7)中, $\#\text{Roots}(x^3 - 1) = 3$ 。□

## 5.2.1 拉格朗日定理

现在我们介绍群论中一个漂亮而重要的定理。

**定义 5.7 陪集(Coset)** 令 $G$ 是一个(阿贝尔)群并且 $H \subseteq G$ 。对于 $a \in G$ ,集合 $a \circ H \stackrel{\text{def}}{=} \{a \circ h \mid h \in H\}$ 称为 $H$ 的一个(左)陪集。

**定理 5.1 拉格朗日定理** 若 $H$ 是 $G$ 的一个子群,则 $\#H \mid \#G$ ,即 $\#H$ 能够整除 $\#G$ 。

**证明** 当 $H = G$ 时, $\#H \mid \#G$ 平凡成立,所以我们考虑 $H \neq G$ 时的情形。

对于任一 $a \in G \setminus H$ ,由封闭律,陪集 $a \circ H$ 是 $G$ 的子集。我们可以证明以下两点:

- i) 对任一 $a \neq a'$ ,如果 $a \notin a' \circ H$ ,那么 $(a \circ H) \cap (a' \circ H) = \emptyset$ 。

① 此处删除了英文版中“所有奇数与0构成的集合也一样”。

ii)  $\#(a \circ H) = \#H$ 。

对于 i), 假设  $\exists b \in (a \circ H) \cap (a' \circ H)$ , 则  $\exists c, c' \in H$ , 使得  $a \circ c = b = a' \circ c'$ 。利用  $H$  中元素的可逆律、单位元律、封闭律和结合律, 我们有

$$a = a \circ e = a \circ (c \circ c^{-1}) = b \circ c^{-1} = (a' \circ c') \circ c^{-1} = a' \circ (c' \circ c^{-1}) \in a' \circ H$$

这与我们的假设  $a \notin a' \circ H$  相矛盾。特别地, 对于  $a \notin H = e \circ H$ , 我们有  $H \cap (a \circ H) = \emptyset$ 。

对于 ii), 由陪集的定义,  $\#(a \circ H) \leq \#H$  平凡成立。假设不等式是严格小于的。这种情况只有当存在  $b \neq c, b, c \in H$ , 使  $a \circ b = a \circ c$  时才有可能成立。利用  $G$  中的可逆律, 得到  $b = c$ , 这与  $b \neq c$  矛盾。

因此,  $G$  被  $H$  划分, 它是  $H$  的互不相交的陪集的并集, 其中每一个陪集中元素个数均为  $\#H$ 。因此,  $\#H \mid \#G$  (通常, 划分一个集合指将该集合分成不相交的子集合)。□

#### 例 5.4

1. 验证例 5.2(3): 对“时钟群” $\mathbb{Z}_{12}$  的每一个子群  $H$ , 均有  $\#H \mid \#\mathbb{Z}_{12}$ 。
2. 在例 5.2(5) 中取  $n = 21$  为例: 我们有  $\text{Fermat}(21) = \{1, 8, 13, 20\}$ , 满足  $\#\text{Fermat}(21) = 4 \mid 12 = \#\mathbb{Z}_{21}^*$ 。□

拉格朗日定理不仅是群论中一个非常漂亮的定理, 而且它还具有很高的应用价值。见 4.4.3.1 节中概率素性检测算法 Prime\_Test, 该算法通过任取  $x \in {}_U \mathbb{Z}_n^*$ , 验证同余式

$$x^{(n-1)/2} \equiv \pm 1 \pmod{n}$$

来判断奇数  $n$  是否为素数。在例 5.2(5) 中, 我们已经看到  $\text{Fermat}(n)$  是由这个同余式所定义子群,  $\text{Fermat}(n)$  是  $\mathbb{Z}_n^*$  的一个真子群, 当且仅当  $n$  不是素数时。所以由拉格朗日定理,  $\#\text{Fermat}(n) \mid \#\mathbb{Z}_n^*$ 。因此, 如果  $n$  不是素数,  $\#\text{Fermat}(n)$  最多是  $\#\mathbb{Z}_n^*$  的二分之一。这告诉我们, 测试的每一步的误差概率上限为  $\frac{1}{2}$ , 这就是 Prime\_Test 的操作原理 (概率空间为  $\mathbb{Z}_n^*$ )。

在 5.2.2 节中, 我们将讨论拉格朗日定理在公钥密码学中的另一个重要应用。

**定义 5.8、商群 (Quotient Group)** 令  $G$  是一个 (阿贝尔) 群, 且  $H \subseteq G$ , 则所有陪集  $a \circ H$  所构成的集合, 称为  $G$  模  $H$  的商群, 其中  $a$  取自于  $G$ , 记为  $G/H$ , 群运算  $*$  定义为  $(a \circ H) * (b \circ H) = (a \circ b) \circ H$ , 单位元是  $e \circ H$ 。

**例 5.5** 令  $n > 0$  是一个整数, 在整数加法运算下, 集合  $n\mathbb{Z} = \{0, \pm n, \pm 2n, \dots\}$  显然是  $\mathbb{Z}$  的一个子群。那么商群

$$\mathbb{Z}/n\mathbb{Z} = \{x + n\mathbb{Z} \mid x \in \mathbb{Z}\}$$

只能有  $n$  个元素。这是因为  $n + n\mathbb{Z} = 0 + n\mathbb{Z}$ ,  $n+1 + n\mathbb{Z} = 1 + n\mathbb{Z}$ , 等等。因此,

$$\mathbb{Z}/(n\mathbb{Z}) = \{0 + n\mathbb{Z}, 1 + n\mathbb{Z}, 2 + n\mathbb{Z}, \dots, n-1 + n\mathbb{Z}\}。$$

将  $n\mathbb{Z}$  视为只包含零模  $n$ , 我们能够得到

$$\mathbb{Z}/n\mathbb{Z} = \mathbb{Z}_n$$

□

事实上,  $\mathbb{Z}/n\mathbb{Z}$  是  $\mathbb{Z}_n$  的正式和标准的记法, 但为表达方便起见, 本书我们总是采用简化记法  $\mathbb{Z}_n$  来代替  $\mathbb{Z}/n\mathbb{Z}$ 。

推论 5.1 令  $G$  是一个有限(阿贝尔)群, 且  $H \subseteq G$ , 则

$$\#(G/H) = \frac{\#G}{\#H} \quad \square$$

例 5.6 令  $m, n$  是正整数, 满足  $m \mid n$ 。参照例 5.5, 我们有

1.  $m\mathbb{Z}_n = \{0, m, 2m, \dots, \lfloor \frac{n-1}{m} \rfloor \cdot m\}$  是  $(\mathbb{Z}_n, +)$  的一个含有  $n/m$  个元素的子集;

2.  $\mathbb{Z}_n/m\mathbb{Z}_n = \mathbb{Z}_m$ ;

3.  $\#(\mathbb{Z}_n/m\mathbb{Z}_n) = \#\mathbb{Z}_m = m = \frac{n}{n/m} = \frac{\#\mathbb{Z}_n}{\#(m\mathbb{Z}_n)}$ 。

例如, 考虑“时钟群” $\mathbb{Z}_{12}$  (即  $n=12$ ) 和它的子群  $3\mathbb{Z}_{12} = \{0, 3, 6, 9\}$  (即  $m=3$ )。与例 5.5 类

似, 读者可以验证  $\mathbb{Z}_{12}/3\mathbb{Z}_{12} = \mathbb{Z}_3$ 。因此,  $\#(\mathbb{Z}_{12}/3\mathbb{Z}_{12}) = \#\mathbb{Z}_3 = 3 = 12/4 = \frac{\#\mathbb{Z}_{12}}{\#(3\mathbb{Z}_{12})}$ 。读者也

可以验证  $m \mid 12$  的所有其他情况。  $\square$

## 5.2.2 群元素的阶

如果我们说在群里, 单位元由于它的惟一性显得有些特别的话, 那么其他的元素也有一些特殊的性质, 其中之一就是与单位元的“距离”。

定义 5.9 群元素的阶 令  $G$  为一群。且  $a \in G$ , 则满足  $a^i = e$  的最小正整数  $i \in \mathbb{N}$  称为元素  $a$  的阶, 记为  $\text{ord}(a)$ 。如果这样的整数  $i$  不存在, 则  $a$  称为无限阶元。

我们要提醒读者注意简化记法  $a^i$  的含义, 这里  $i$  是一个整数,  $a$  是一个群元素。这种简化记法的含义在定义 5.4 中已经给出, 并且在注释 5.1 中已经做了进一步解释。

例 5.7

1. 在“时钟群” $\mathbb{Z}_{12}$  中, 因为 12 是满足  $12 \cdot 1 \equiv 0 \pmod{12}$  的最小正整数, 所以  $\text{ord}(1) = 12$ ; 读者可以验证:  $\text{ord}(2) = 6, \text{ord}(3) = 4, \text{ord}(4) = 3, \text{ord}(5) = 12$ , 并设法找出其余元素的阶。
2. 在例 5.1(6)的  $B$  中,  $\text{ord}(F) = 1, \text{ord}(T) = 2$ 。
3. 在例 5.1(7)的  $\text{Roots}(x^3 - 1)$  中,  $\text{ord}(\alpha) = \text{ord}(\beta) = 3, \text{ord}(1) = 1$ 。
4. 在  $\mathbb{Z}$  中,  $\text{ord}(1) = \infty$ 。  $\square$

推论 5.2 拉格朗日 令  $G$  为一有限群, 且  $a \in G$  为任一元, 那么  $\text{ord}(a) \mid \#G$ 。

证明 取任一  $a \in G$ , 如果  $a = e$ , 那么  $\text{ord}(a) = 1$ , 所以  $\text{ord}(a) \mid \#G$  是平凡的情况。令  $a \neq e$ , 因为  $G$  为有限群, 我们有  $1 < \text{ord}(a) < \infty$ 。元素

$$a, a^2, \dots, a^{\text{ord}(a)} = e \quad (5.2.1)$$

一定是互不相同的, 否则, 存在非负整数  $r$  和  $s$ , 满足  $1 \leq r < s \leq \text{ord}(a)$ , 使得  $a^r = a^s$ 。两边同时运用  $(a^r)^{-1}$  的“可逆律”, 得到  $a^{s-r} = e$ , 这里  $0 < s-r < \text{ord}(a)$ 。这与  $\text{ord}(a)$  是满足  $a^{\text{ord}(a)} = e$  的最小正整数的定义相矛盾。

容易验证(5.2.1)式中的  $\text{ord}(a)$  个元素构成  $G$  的一个子群, 由拉格朗日定理, 有  $\text{ord}(a) \mid \#G$ 。  $\square$

作为拉格朗日定理的一个直接应用, 推论 5.2 给我们提供了群的阶和群中元素的阶之间的关系, 这个关系在公钥密码学中有重要的应用: 著名的 Rivest、Shamir 和 Adleman (RSA) 密码

体制是在有秘密阶的群中实现的,该群的阶只有密钥的拥有者才知道,密文可以看做是群中的任意元素。有了群阶的知识,密钥的拥有者就能够利用元素的阶和群阶之间的关系将密文变换为明文(即解密)。我们将在 8.5 节中讨论 RSA 密码体制。

### 5.2.3 循环群

例 5.1(4)表明我们可以方便地将  $\mathbb{Z}_n$  看做是一个圆分割成的  $n$  个点,这个圆(或是这  $n$  个点)由  $n$  个重复的运算  $a^1, a^2, \dots, a^n$  构成,其中某个元素  $a \in \mathbb{Z}_n$ , 这就是  $\mathbb{Z}_n$  的循环图像。将模  $n$  加法作为群运算,  $a = 1$  给出了  $\mathbb{Z}_n$  的一种循环图像。读者可以验证,在例 5.1(4)中  $n = 12$  的情况下, 5、7、11 是能够给出  $\mathbb{Z}_{12}$  的循环图像的另外三个元素。

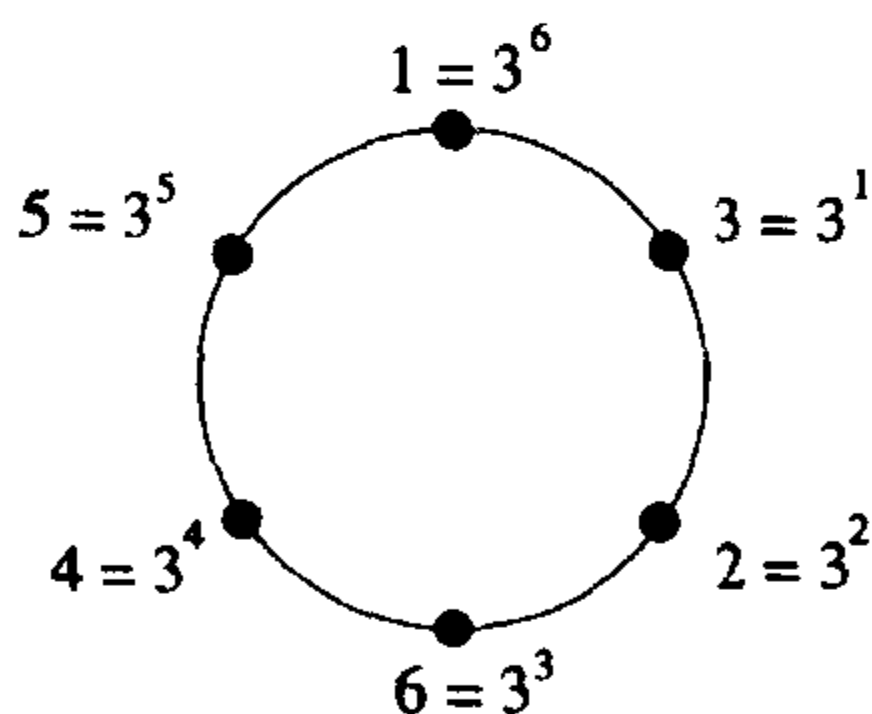
通俗地讲,如果一个群存在一种循环图像,那么我们说这个群就是循环群。循环群是有很好的性质的群,它在密码学中有广泛的应用。

**定义 5.10 循环群、群生成元** 如果存在一个元素  $a \in G$ , 对任一  $b \in G$ , 都存在一个整数  $i \geq 0$ , 使得  $b = a^i$ , 则群  $G$  就称为循环群, 元素  $a$  称为  $G$  的一个生成元,  $G$  也称为由  $a$  生成的群。当一个群由  $a$  生成的时候, 记做  $G = \langle a \rangle$ 。

循环群的生成元也称为这个群的单位元的本原根。这个名称的含义将在 5.4.3 节(定理 5.11)中进行阐述。

#### 例 5.8

1. 对于  $n \geq 1$ , 加法群  $\mathbb{Z}_n$  是循环群, 这是因为 1 显然是一个生成元。
2. 例 5.1(6)中的  $B$  是一个由  $T$  生成的循环群。
3. 例 5.1(7)中的  $\text{Roots}(x^3 - 1)$  是由  $\alpha$  或  $\beta$  生成的循环群。
4. 令  $p$  为一个素数, 则乘法群  $\mathbb{Z}_p^*$  是循环群。这是因为  $\mathbb{Z}_p^*$  包含有阶为  $p - 1 = \# \mathbb{Z}_p^*$  的元素, 因此这个元素生成了整个群。在算法 4.6 中我们已经看到了  $\mathbb{Z}_p^*$  包含一个生成元的非严格证明, 在定理 5.12 中我们将给出  $\mathbb{Z}_p^*$  是一个循环群的严格证明。
5. 在群  $\mathbb{Z}_7^*$  中, 3 是一个生成元, 这个元素给出的  $\mathbb{Z}_7^*$  的一个循环图像(记住群运算是模 7 乘法)如下:



□

**定义 5.11 欧拉函数** 对于  $n \in \mathbb{N}, n \geq 1$ , 整数  $k$  的个数称为欧拉函数  $\phi(n)$ , 这里  $k$  满足  $0 \leq k < n$ , 且  $\gcd(k, n) = 1$ 。

由循环群可以推导出许多有用的结果。



## 定理 5.2

1. 循环群的每一个子群均为循环群。
2. 对  $\# \langle a \rangle$  的每一个正因子  $d$ ,  $\langle a \rangle$  恰包含一个  $d$  阶子群。
3. 如果  $\# \langle a \rangle = m$ , 那么  $\# \langle a^k \rangle = \text{ord}(a^k) = m / \gcd(k, m)$ 。
4. 对  $\# \langle a \rangle$  的每一个正因子  $d$ ,  $\langle a \rangle$  包含  $\phi(d)$  个  $d$  阶元。
5. 令  $\# \langle a \rangle = m$ , 那么  $\langle a \rangle$  包含  $\phi(m)$  个生成元, 这些生成元形如  $a^r$ , 其中  $\gcd(r, m) = 1$ 。

证明

1. 令  $H \subseteq \langle a \rangle$ , 如果  $H = \langle e \rangle$  或  $H = \langle a \rangle$ , 那么  $H$  显然是循环群, 所以我们仅仅考虑  $H$  的其他情形。令  $d > 1$  是使得  $a^d \in H$  的最小正整数, 如果对某个  $s > d$  有  $a^s \in H$ , 用  $d$  整除  $s$ , 得到  $s = dq + r$ , 对于某个  $0 \leq r < d$ 。既然  $a^{dq} \in H$ , 我们有  $a^r = a^{s-dq} \in H$ 。由  $d$  的最小性和  $H \neq \langle a \rangle$  可得  $r = 0$ , 所以  $s$  是  $d$  的倍数, 因此  $H$  仅包含  $a^d$  的幂, 它是一个循环群。
2. 令  $d > 1$  且  $d \mid m = \# \langle a \rangle$ , 因为  $d$  是满足  $(a^{\frac{m}{d}})^d = e$  的最小整数, 所以  $\langle a^{\frac{m}{d}} \rangle$  是  $\langle a \rangle$  的一个  $d$  阶子群。我们假设存在  $\langle a \rangle$  的另一个与  $\langle a^{\frac{m}{d}} \rangle$  不同的  $d$  阶子群, 由 1, 那样的子群必是循环群, 因此假定是  $\langle a^k \rangle$ , 对于某个  $k > 1$ 。由  $a^{kd} = e$  及  $m$  的最小性, 我们有  $m \mid kd$ , 或者等价地有  $\frac{m}{d} \mid k$ 。所以  $a^k \in \langle a^{\frac{m}{d}} \rangle$ , 即  $\langle a^k \rangle \subseteq \langle a^{\frac{m}{d}} \rangle$ , 又由于这两个群具有相同的阶, 所以  $\langle a^k \rangle = \langle a^{\frac{m}{d}} \rangle$ 。这与我们的假设  $\langle a^k \rangle \neq \langle a^{\frac{m}{d}} \rangle$  相矛盾。
3. 令  $d = \gcd(k, m)$ , 那么由 2,  $\langle a \rangle$  有惟一的  $d$  阶子群, 令这个子群是  $\langle a^\ell \rangle$ , 对于某个最小的  $\ell > 1$ , 即  $\ell$  是满足  $a^{d\ell} = e$  的最小整数, 由  $m$  的最小性, 我们有  $m \mid d\ell$ , 或者等价地有  $\frac{m}{d} \mid \ell$ 。由  $\ell$  的最小性, 所以  $d = \gcd(\ell, m)$ , 即  $\ell = k$ 。
4. 令  $d \mid m = \# \langle a \rangle$ ,  $a^k$  是  $\langle a \rangle$  中的任一元,  $0 \leq k < m$ 。由 3, 元素  $a^k$  阶为  $d$  当且仅当  $\frac{m}{d} = \gcd(k, m)$ , 记  $k = c \frac{m}{d}$ , 其中  $0 \leq c < d$ 。则  $\gcd(k, m) = \frac{m}{d}$  等价于  $\gcd(c, d) = 1$ , 故由定义 5.11, 共有  $\phi(d)$  个那样的  $c$ 。
5. 对于  $m = \# \langle a \rangle$ , 由 4,  $\langle a \rangle$  包含  $\phi(m)$  个  $m$  阶元素, 又由于它们都是  $m$  阶元素, 因此均是  $\langle a \rangle$  的生成元。再由 3, 这些生成元形如  $a^r$ , 其中  $\gcd(r, m) = 1$ 。□

推论 5.3 素阶群均为循环群, 且该群中任意非单位元均为生成元。

证明 令  $G$  为一素数  $p$  阶群,  $a \in G$  为任一非单位元, 由推论 5.2 可知,  $\text{ord}(a) \mid \# G = p$ 。由于  $a \neq e$ , 故  $\text{ord}(a) \neq 1$ , 所以  $\text{ord}(a) = p$ 。因此  $\langle a \rangle = G$ , 即  $a$  是  $G$  的一个生成元。□

例 5.9 考虑“时钟群” $\mathbb{Z}_{12}$ , 它是循环群:

- 由于  $1 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 1 阶子群  $\{0\}$ ; 因为  $\phi(1) = 1$ , 所以 1 阶元只有 0;
- 由于  $2 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 2 阶子群  $\{0, 6\}$ ; 因为  $\phi(2) = 1$ , 所以 2 阶元只有 6;
- 由于  $3 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 3 阶子群  $\{0, 4, 8\}$ ; 因为  $\phi(3) = 2$ , 所以 3 阶元有 2 个, 为 4 和 8;
- 由于  $4 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 4 阶子群  $\{0, 3, 6, 9\}$ ; 因为  $\phi(4) = 2$ , 所以 4 阶元有 2 个, 为 3 和 9;
- 由于  $6 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 6 阶子群  $\{0, 2, 4, 6, 8, 10\}$ ; 因为  $\phi(6) = 2$ , 所以 6 阶元有 2 个, 为 2 和 10;

- 由于  $12 \mid 12$ , 故  $\mathbb{Z}_{12}$  包含一个 12 阶子群  $\mathbb{Z}_{12}$ ; 因为  $\phi(12) = 4$ , 所以 12 阶元有 4 个, 为 1、5、7 和 11。

读者可以类似地分析乘法群  $\mathbb{Z}_7^*$ 。

□

#### 5.2.4 乘法群 $\mathbb{Z}_n^*$

令  $n = pq$ , 其中  $p$  和  $q$  为不同的奇素数。乘法群  $\mathbb{Z}_n^*$  在现代密码学中是非常重要的。现在让我们来看看它的结构。在本节中我们规定  $n$  均为那样的一个合数。

由于  $\mathbb{Z}_n^*$  中的元素均是小于  $n$  且与  $n$  互素的正整数, 由定义 5.11 可知, 该群包含  $\phi(n) = (p-1)(q-1)$  个元素 [引理 6.1 保证了  $\phi(n) = (p-1)(q-1)$ ]。

**定理 5.3**  $\mathbb{Z}_n^*$  中任一元素的阶都能整除  $\text{lcm}(p-1, q-1)$ 。

**证明** 令  $a \in \mathbb{Z}_n^*$ , 由费马小定理 (6.4 节中的定理 6.10), 我们知道

$$a^{(p-1)} \equiv 1 \pmod{p}$$

记  $\lambda = \text{lcm}(p-1, q-1)$ , 平凡地有

$$a^\lambda \equiv 1 \pmod{p}$$

对称地, 我们也能导出

$$a^\lambda \equiv 1 \pmod{q}$$

这两个同余式实际上说明了  $a^\lambda - 1$  既是  $p$  的倍数也是  $q$  的倍数。既然  $p$  和  $q$  是不同的素数, 故  $a^\lambda - 1$  一定是  $n = pq$  的倍数。即

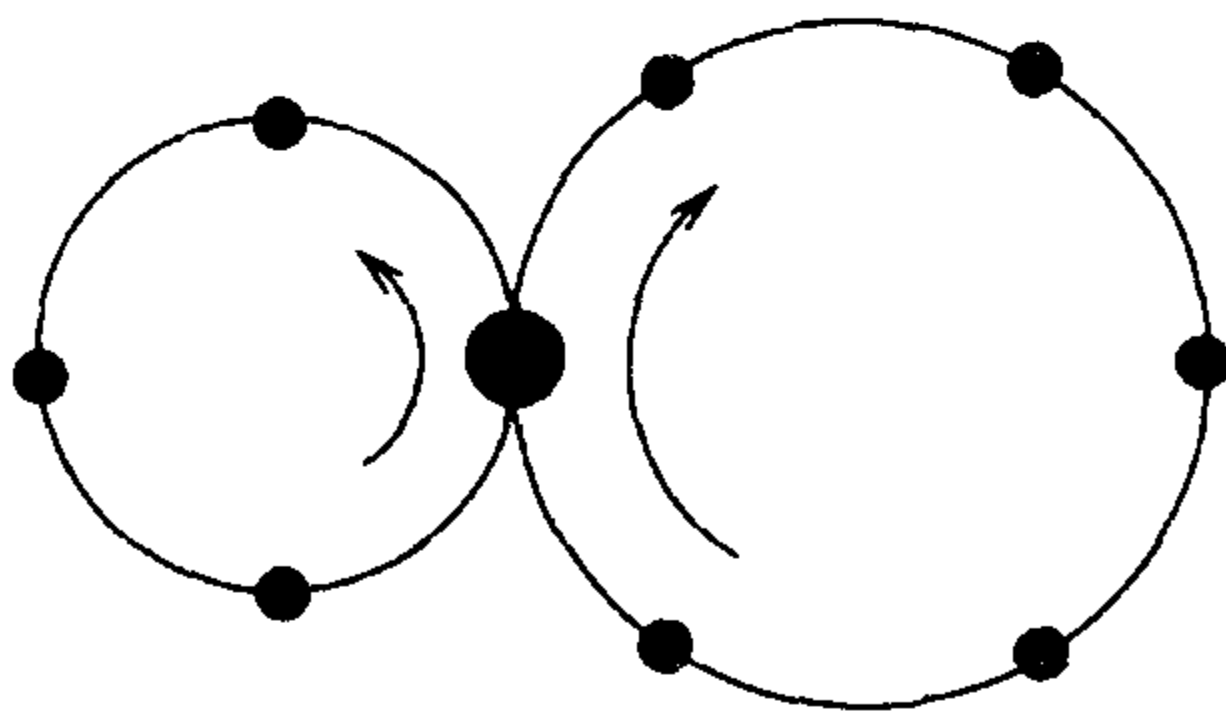
$$a^\lambda \equiv 1 \pmod{n}$$

因此,  $\lambda$  是  $a$  模  $n$  的阶的倍数。

□

注意到  $(p-1)$  和  $(q-1)$  都是偶数, 因此  $\lambda = \text{lcm}(p-1, q-1) < (p-1)(q-1) = \phi(n)$ 。定理 5.3 告诉我们在  $\mathbb{Z}_n^*$  中没有  $\phi(n)$  阶元素, 即  $\mathbb{Z}_n^*$  不包含生成元。故由定义 5.10 可知,  $\mathbb{Z}_n^*$  不是循环群。数值  $\lambda(n)$  称为  $n$  的 **Carmichael 数**。

**例 5.10** 对  $n = 5 \times 7 = 35$ , 令  $a \in \mathbb{Z}_{35}^*$  是这样的一个元: (i)  $a \pmod{5} \in \mathbb{Z}_5^*$  有最大的阶 4, 因此它给出了循环群  $\mathbb{Z}_5^*$  的一个循环图像 (见下图中左边的圆, 该圆周期为 4); (ii)  $a \pmod{7} \in \mathbb{Z}_7^*$  有最大的阶 6, 因此它给出了循环群  $\mathbb{Z}_7^*$  的一个循环图像 (见下图中右边的圆, 该圆周期为 6)。



因此, 可以将  $a \in \mathbb{Z}_{35}^*$  的阶看做是由两个啮合的齿轮所决定的周期, 其中一个齿轮有 4 个齿, 另一个有 6 个齿。开始我们在两个轮子的啮合处标识一个大点 (见上图)。现在让啮合的齿轮旋转, 大点分离成了两个轮子上的两个点, 在四个齿的轮子旋转了 3 周、六个齿

的轮子旋转了2周之后,这两个分离点又重合。因此, $a \in \mathbb{Z}_{35}^*$ 的阶(周期)恰是大点从分离到重合之间的距离,也就是 $3 \times 4 = 2 \times 6 = 12 = \text{lcm}((5-1), (7-1))$ 。□

用 $\text{ord}_n(a)$ 表示一个元素模正整数 $n$ 的阶。通常,任一元素 $a \in \mathbb{Z}_n^*$ 的阶 $\text{ord}_n(a)$ 可由 $\text{ord}_p(a)$ 及 $\text{ord}_q(a)$ 如下定义:

$$\text{ord}_n(a) = \text{lcm}(\text{ord}_p(a), \text{ord}_q(a)) \quad (5.2.2)$$

既然 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ 都是循环群,它们分别有最大阶为 $p-1$ 和最大阶为 $q-1$ 的元素,因此, $\mathbb{Z}_n^*$ 含有最大阶为 $\text{lcm}(p-1, q-1)$ 的元素;另一方面,某一最大阶元 $a \in \mathbb{Z}_n^*$ 满足 $\text{ord}_p(a) < p-1$ 和/或 $\text{ord}_q(a) < q-1$ 。例如,由于 $\text{lcm}(4, 3) = \text{lcm}(4, 6)$ 且包含3阶元,所以群 $\mathbb{Z}_{35}^*$ 包含最大周期为12的元素,该元素可由两个啮合的齿轮来表示,这两个齿轮一个是4个齿的,另一个是3个齿的。

下一章我们将给出 $\mathbb{Z}_n^*$ 中元素与 $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ 中元素对之间的一个一一映射。通过该映射,给出一个原像,就可以求出一个像点。因此该映射给出了一种由循环群 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ 中的元素构造 $\mathbb{Z}_n^*$ 中元素的方法。因为 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ (循环群)具有很好的性质,所以 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ 中元素的运算通常是更容易的。例如,由于计算 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ 中元素的平方根比较容易,我们可以利用映射通过计算 $\mathbb{Z}_p^*$ 和 $\mathbb{Z}_q^*$ 中元素的平方根来构造 $\mathbb{Z}_n^*$ 中元素的平方根。

### 5.3 环和域

直到有一天古代牧羊人定居下来,成为一个农夫,他需要与邻人计算土地的面积。由牧羊人变成的农夫使他们逐渐意识到,使用一种基本的运算来处理每件事情行不通了:他们不仅要求和,而且还需要进行乘积运算。从此开始了一个对象集上有两种运算的需求。

**定义 5.12 环(Ring)** 一个同时有两种运算:(加法) $+$ 和(乘法) $\cdot$ 的集合,如果满足如下性质,就称为环 $R$ :

1.  $R$ 在加法 $+$ 下是一个阿贝尔群,加法单位元记做 $0$ (称为零元);
2.  $R$ 在乘法 $\cdot$ 下满足封闭律、结合律和单位元律,乘法单位元记做 $1$ (称为单位元), $1 \neq 0$ ;
3.  $\forall a, b \in R$ ,有 $a \cdot b = b \cdot a$ ; (交换律)
4.  $\forall a, b, c \in R$ ,有 $a \cdot (b + c) = a \cdot b + a \cdot c$ 。 (分配律)

在这个定义中, $0$ 和 $1$ 使用粗体是为了强调这两个元素是抽象元,不同于整数中的 $0$ 和 $1$ [例如很快就要看到的例5.11(3)]。

类似于我们所做的交换群的规定,在定义5.12中我们规定乘法满足交换律,所以定义5.12定义了一个交换环,也就是本书要考虑的环。我们也要强调 $+$ 和 $\cdot$ 是抽象运算。也就是说,它们未必是整数中通常的加法和乘法。在不引起混淆的情况下,我们将 $a \cdot b$ 缩写为 $ab$ ,而只有在没有运算数的情况下,才会明确写出运算“ $\cdot$ ”。

#### 例 5.11 环

1. 在通常的加法和乘法运算下, $\mathbb{Z}$ 、 $\mathbb{Q}$ 、 $\mathbb{R}$ 和 $\mathbb{C}$ 均是环, $0=0, 1=1$ 。
2. 对任意 $n > 0$ ,在模 $n$ 加法和模 $n$ 乘法运算下, $\mathbb{Z}_n$ 是一个环, $0=0, 1=1$ 。
3. 令 $B$ 为例5.1(6)中定义加法群,零元为 $F$ 。令乘法运算是 $\wedge$ (逻辑与): $F \wedge F = F$ ,  
 $F \wedge T = T \wedge F = F, T \wedge T = T$ 。则 $B$ 是一个有单位元 $T$ 的环。 □

初看起来,定义 5.12 仅仅定义了非零元间的乘法。实际上,由分配律也定义了零元和其他元素之间的乘法。例如, $0a = (b + (-b))a = ba + (-b)a = ba - ba = 0$ 。而且,环可以有零因子,即满足以下条件的元素  $a$  和  $b$ :  $ab = 0$ , 且  $a \neq 0, b \neq 0$ 。例如,设  $n = k\ell$  是  $n$  的一个非平凡分解,  $k$  和  $\ell$  均是环  $\mathbb{Z}_n$  中的非零元,而乘积  $k\ell = n = 0 \pmod{n}$  是零元。

**定义 5.13 域(Field)** 如果一个环中的非零元在乘法运算下构成群,则该环就称为域。

域中乘法群(即非零元)满足封闭律,这表明域  $F$  不含零因子,即对任意  $a, b \in F, ab = 0$  可推出  $a = 0$  或  $b = 0$ 。

### 例 5.12 域

1. 在通常的加法和乘法运算下,  $\mathbb{Q}, \mathbb{R}$  和  $\mathbb{C}$  均是域,  $0 = 0, 1 = 1$ 。
2. 例 5.11(3)中的二元环  $B$  是一个域。
3. 令  $p$  为素数,在模  $p$  加法和模  $p$  乘法运算下,  $\mathbb{Z}_p$  是一个域,  $0 = 0, 1 = 1$ 。 □

我们即将看到更多域的例子。

注意到在通常的整数加法和乘法运算下,  $\mathbb{Z}$  不是域,这是因为任何非零元在  $\mathbb{Z}$  中都没有乘法逆(破坏了可逆律)。当  $n$  为合数时,如我们所见,  $\mathbb{Z}_n$  包含零因子(破坏了封闭律),因此,  $\mathbb{Z}_n$  也不是域。

有时候群、环和域间的不同对我们并没有影响,在这种情况下,我们用代数结构来指代这三种结构中的任一种。

有限群、子群、商群和群的阶的概念可以直接推广到环和域。

**定义 5.14** 一个代数结构如果包含有限个元素,就说该代数结构是有限的。元素的个数称为这个结构的阶。

如果一个代数结构  $A$  的一个非空子集  $S$  在  $A$  的运算( $s$ )下自身成为一个代数结构,那么  $S$  就称为代数结构  $A$  的子结构。如果  $S \neq A$ ,那么  $S$  就称为代数结构  $A$  的真子结构。

令  $A$  是一个代数结构,  $B \subseteq A$  是  $A$  的一个子结构,则所有陪集  $a \circ B$  所构成的集合称为  $A$  模  $B$  的商结构,其中  $a$  遍历  $A$ ,记做  $A/B$ ,运算  $*$  定义为  $(a \circ B) * (b \circ B) = (a \circ b) \circ B$ ,单位元是  $0 \circ B$  和  $1 \circ B$ 。

从定义 5.14 可知,环(或域)不仅可以有子环(或子域),也可以有子群(或子环和子群)。我们将在 5.4 节中看到这样的例子。

## 5.4 有限域的结构

有限域在密码学和密码协议中有着广泛的应用。在公钥密码学中,Diffie 和 Hellman 的开创性工作和 Diffie-Hellman 密钥交换协议[99](见 8.3 节),最初都是在有特殊形式的有限域中提出来的。由于 Diffie 和 Hellman 所做的工作,无数的基于有限域的密码体制和协议已经被提出:ElGamal 密码体制[103]、Schnorr 身份协议和签名方案[259]、Chaum 的零知识不可否认签名和 Chaum 与 Pedersen 的零知识证明协议[74]都是著名的例子。一些新的密码体制,如高级加密标准[221](见 7.7 节)和 XTR 密码体制[177]都是在更一般形式的有限域上实现的。有限域也是椭圆曲线的基础,而椭圆曲线构成了一类密码体制的基础(例如[168])。

现在让我们对有限域的结构进行一下完整的讨论。

### 5.4.1 含有素数个元素的有限域

有最简单结构的有限域就是阶(元素的个数)为素数的有限域。然而,这样的域在密码学中的应用却最广泛。

**定义 5.15 素域** 不含真子域的域称为素域。

例如, $\mathbb{Q}$ 为素域,而由于 $\mathbb{Q}$ 为 $\mathbb{R}$ 的真子域,所以 $\mathbb{R}$ 不是素域, $\mathbb{Q}$ 是一个无限域。在有限域中,我们很快将看到素域必包含素数个元素,即素域必须有素数阶。

**定义 5.16 同态和同构** 令  $A, B$  是两个代数结构,如果映射  $f: A \mapsto B$  保持  $A$  的运算,即如果  $\circ$  是  $A$  中的运算,  $*$  是  $B$  中的运算,那么  $\forall x, y \in A$ , 有  $f(x \circ y) = f(x) * f(y)$ , 则该映射就称为  $A$  到  $B$  的同态。如果  $f$  是  $A$  到  $B$  上的一一同态,那么  $f$  就称为一个同构,我们就说  $A$  和  $B$  是同构的。

如果  $f: A \mapsto B$  是一个同态,且  $e$  是  $A$  中的一个单位元(加法的或乘法的),那么

$$f(e) * f(e) = f(e \circ e) = f(e)$$

所以  $f(e)$  是  $B$  中单位元,而且对任一  $a \in A$ , 有

$$f(a) * f(a^{-1}) = f(a \circ a^{-1}) = f(e)$$

因此对所有  $a \in A$ , 都有  $f(a^{-1}) = f(a)^{-1}$ 。并且如果映射是一一映射(即  $A$  和  $B$  是同构的),那么  $A$  和  $B$  有同样的元素个数。两个同构的代数结构视为具有同样的结构。

#### 例 5.13 同构的代数结构

- i) 集合  $\{0, 1\}$  中的运算分别是整数模 2 加法  $+$  和整数乘法  $\cdot$ , 将它记为  $\mathbb{F}_2$ 。对于例 5.12(2)中的域  $B$ , 按照验证程序, 映射  $f(0) = F, f(1) = T$  是一个同构映射, 所以  $\mathbb{F}_2$  同构于  $B$ , 因此  $\mathbb{F}_2$  一定是一个域。
- ii) 对于任意素数  $p$ , 按照验证程序, 函数  $f(x) = g^x \pmod{p}$  是加法群  $\mathbb{Z}_{p-1}$  和乘法群  $\mathbb{Z}_p^*$  之间的同构映射, 因此这两个群是同构的。  $\square$

显然,所有的二元域彼此同构,因此都同构于 $\mathbb{F}_2$ 。二元域是最简单的域:它包含两个必有的元素,即零元和单位元,除此之外不包含其他元。由于在同构作用下,没有必要区分这些域的不同,因此,我们将 $\mathbb{F}_2$ 视做惟一的阶为2的域。

**例 5.14 素数阶的有限域** 令  $p$  为任意素数,则 $\mathbb{Z}_p$ (整数模  $p$ )是一个  $p$  阶有限域(即含有  $p$  个元素),其中域运算为模  $p$  加法和模  $p$  乘法。事实上,我们已经在例 5.11(2)中表明了 $\mathbb{Z}_p$ 是一个加环,在例 5.1(5)中已经说明 $\mathbb{Z}_p$ 的非零元(记为 $\mathbb{Z}_p^*$ )构成一个乘法群。  $\square$

**定义 5.17 域  $\mathbb{F}_p$**  令  $p$  为一个素数,有限域 $\mathbb{Z}_p$ 记为 $\mathbb{F}_p$ 。

令  $F$  为任一素数  $p$  阶的有限域,由于我们可以构造  $F$  到  $\mathbb{F}_p$  的一一映射(即映射是同构的),所以任意  $p$  阶有限域都同构于 $\mathbb{F}_p$ 。由于我们没有必要区分彼此同构的域,因此能够无碍地将  $p$  阶有限域称为 $\mathbb{F}_p$ 。

令  $A$  是一个具有加法运算“+”的有限代数结构,  $a$  是  $A$  中任一非零元。观察下面的序列:

$$a, 2a = a + a, 3a = a + a + a, \dots \quad (5.4.1)$$

因为  $A$  有限, 所以元素  $a$  有有限阶, 因此在这个序列里一定存在元素对  $(ia, ja)$ , 其中  $i < j$  是整数, 满足  $ja - ia = (j - i)a = 0$ 。

我们要提醒读者注意, 定义 5.4 和注释 5.1 中乘法的写法  $ia$  的缩写含义, 这里  $i$  是整数,  $a$  是一个抽象元素。

**定义 5.18 代数结构的特征**  $A$  是一个代数结构, 对每一个  $a \in A$ , 满足  $na = 0$  的最小正整数  $n$  称为  $A$  的特征, 记为  $\text{char}(A)$ 。如果这样的正整数  $n$  不存在, 就说  $A$  的特征为 0。

**定理 5.4** 每一个有限域的特征均为素数。

**证明** 令  $F$  为有限域,  $a \in F$  为任意非零元, 由式(5.4.1)中序列导出的  $(j - i)a = 0$ , 其中  $j > i$ , 可知  $F$  一定有正特征, 设为  $n$ 。由于  $F$  至少有两个元(即零元和单位元), 所以  $n \geq 2$ 。如果  $n > 2$  不是素数, 我们可写为  $n = k\ell$ , 其中  $k, \ell \in \mathbb{Z}, 1 < k, \ell < n$ 。那么

$$0 = n1 = (k\ell)1 = (k\ell)11 = (k1)(\ell 1)$$

由于  $F$  的非零元构成乘法群(不包含 0), 所以表明了  $k1 = 0$  或  $\ell 1 = 0$ , 因此对所有的  $a \in F$ ,  $ka1 = (k1)a = 0$  或  $\ell a1 = (\ell 1)a = 0$ , 这与特征  $n$  的定义相矛盾。□

## 5.4.2 模不可约多项式的有限域

有限素域的阶等于域的特征。然而这并不是有限域的一般情况, 可用多项式来构造有限域的更一般的形式。

### 5.4.2.1 代数结构上的多项式

在第 4 章中我们已经讨论了整数上的多项式, 现在让我们熟悉一下抽象代数结构上的多项式。

**定义 5.19 代数结构上的多项式** 令  $A$  为具有加法和乘法的代数结构, 形如  $f(x) = \sum_{i=0}^n a_i x^i$  的表达式称为  $A$  上的多项式, 其中  $n$  为非负整数, 系数  $a_i, 0 \leq i \leq n$  是  $A$  中的元素,  $x$  是不属于  $A$  的一个符号。系数  $a_n$  称为首系数, 当  $n \neq 0$  时,  $a_n$  不是  $A$  中的零元; 整数  $n$  称为  $f(x)$  的次数, 记为  $n = \deg(f(x)) = \deg(f)$ ; 如果首系数是  $a_0$ , 那么  $f$  称为常数多项式; 如果首系数是  $a_0 = 0$ , 那么  $f$  称为零多项式, 记为  $f = 0$ ; 我们将代数结构  $A$  上的所有多项式集合记为  $A[x]$ 。

对于  $f, g \in A[x], f(x) = \sum_{i=0}^n a_i x^i$  和  $g(x) = \sum_{i=0}^m b_i x^i$ , 我们有

$$f(x) + g(x) = \sum_{i=0}^{\max(n, m)} c_i x^i, \text{ 其中 } c_i = \begin{cases} a_i + b_i & i = 0, 1, \dots, \min(n, m) \\ a_i & i = m + 1, \dots, n \\ b_i & i = n + 1, \dots, m \end{cases} \quad (5.4.2)$$

和



$$f(x)g(x) = \sum_{k=0}^{n+m} c_k x^k, \text{ 其中 } c_k = \sum_{\substack{i+j=k \\ 0 \leq i \leq n \\ 0 \leq j \leq m}} a_i b_j \quad (5.4.3)$$

很容易看出,如果  $A$  是一个环,那么  $A[x]$  也是一个环,而  $A$  是  $A[x]$  的子环。由环上的多项式间的加法和乘法可以得出多项式次数间的下列关系式:

$$\deg(f+g) \leq \max(\deg(f), \deg(g)),$$

$$\deg(fg) \leq \deg(f) + \deg(g)$$

如果  $A$  是一个域,那么因为域没有零因子,如果  $a_n \neq 0$  且  $b_m \neq 0$ ,我们就有  $c_{n+m} = a_n b_m \neq 0$ ,故如果  $A$  是一个域,则

$$\deg(fg) = \deg(f) + \deg(g)$$

令  $f, g \in A[x]$  且  $g \neq 0$ ,类似于整数间除法的情形(见 4.3.2.1 节),我们总是能写成

$$f = gq + r, \quad q, r \in A[x] \text{ 且 } \deg(r) < \deg(g) \quad (5.4.4)$$

**例 5.15** 考虑  $f(x) = x^5 + x^4 + x^3 + x^2 + x + 1 \in \mathbb{F}_2[x]$ ,  $g(x) = x^3 + x + 1 \in \mathbb{F}_2[x]$ 。通过长除法可以计算  $q, r \in \mathbb{F}_2[x]$

$$\begin{array}{r} x^3 + x + 1 \overline{) \begin{array}{cccccccc} & x^2 & & + & x & & & \\ x^5 & + & x^4 & + & x^3 & + & x^2 & + & x & + & 1 \\ \underline{x^5} & & & & + & x^3 & + & x^2 & & & \\ & x^4 & & & & & + & x & + & 1 \\ & \underline{x^4} & & & & + & x^2 & + & x & & \\ & & & & & x^2 & & + & 1 \end{array}} \end{array}$$

因此,  $q = x^2 + x, r = x^2 + 1$ 。 □

**定义 5.20 不可约多项式** 令  $A$  是一个代数结构,多项式  $f \in A[x]$ ,如果  $f$  有一个正次数,且由  $f = gh, g, h \in A[x]$  可推导出或者  $g$  或者  $h$  是常数多项式,那么就说  $f$  在  $A$  上是不可约的(或在  $A[x]$  上是不可约的,或在  $A[x]$  上是素的)。如果多项式  $f$  在  $A$  上不是不可约的,那么就说它在  $A$  上是可约的。

注意到多项式的可约性依赖于该多项式定义在什么样的代数结构上。一个多项式在一种代数结构上是可约的,但在另一种代数结构上可能就是不可约的。

**例 5.16** 对于二次多项式  $f(x) = x^2 - 2x + 2$ : (i) 在通常的无限代数结构上讨论它的可约性; (ii) 对于任意奇素数  $p$ ,在有限域  $\mathbb{F}_p$  上讨论它的可约性; (iii) 对  $p < 10$ ,在  $\mathbb{F}_p$  上分解  $f(x)$ 。利用初等代数中的求根公式,我们计算  $f(x) = 0$  的两个根为

$$\alpha = 1 + \sqrt{-1}, \beta = 1 - \sqrt{-1}$$

i) 由于  $\sqrt{-1}$  不属于  $\mathbb{R}$ , 因此  $f(x)$  在  $\mathbb{R}$  上不可约(因此在  $\mathbb{Z}$  或  $\mathbb{Q}$  上也不可约),但是因为  $\sqrt{-1} = i \in \mathbb{C}$ , 因此  $f(x)$  在  $\mathbb{C}$  上可约:

$$f(x) = (x - 1 - i)(x - 1 + i)$$

ii) 显然,对于任意奇素数  $p$ ,  $f(x)$  在  $\mathbb{F}_p$  上是可约的,当且仅当  $\sqrt{-1}$  是  $\mathbb{F}_p$  中的一个元素,或者等价地说,  $-1$  是模  $p$  的一个平方数。

数  $x$  称为模  $p$  的平方数, 当且仅当存在  $y(\bmod p)$ , 满足  $y^2 \equiv x(\bmod p)$ 。由费马小定理 (6.4 节中的定理 6.10), 我们知道所有的  $x(\bmod p)$  满足  $x^{p-1} \equiv 1(\bmod p)$ 。当  $p$  是一个奇素数时, 费马小定理等价于

$$x^{\frac{p-1}{2}} \equiv \pm 1(\bmod p) \quad (5.4.5)$$

对于所有的  $x, 0 < x < p$  (这里  $-1$  表示  $p-1$ )。如果  $x$  是模  $p$  的一个平方数, 那么式 (5.4.5) 成为

$$x^{\frac{p-1}{2}} \equiv (y^2)^{\frac{p-1}{2}} \equiv y^{p-1} \equiv 1(\bmod p)。$$

因此, 我们知道式 (5.4.5) 给出了一个判定  $x$  是否为模一个奇素数  $p$  的平方数的准则标准: 如果测试结果等于 1 (或  $-1$ ), 那么  $x$  是模  $p$  的一个平方数 (或非平方数)。

我们最终得出的结论是, 对任意奇素数  $p, f(x)$  在  $\mathbb{F}_p$  上是可约的, 当且仅当  $(-1)^{\frac{p-1}{2}} \equiv 1(\bmod p)$ ;  $f(x)$  在  $\mathbb{F}_p$  上是不可约的, 当且仅当  $(-1)^{\frac{p-1}{2}} \equiv -1(\bmod p)$ 。换句话说, 如果  $p \equiv 1(\bmod 4)$  (或  $p \equiv 3(\bmod 4)$ ), 那么  $f(x)$  在  $\mathbb{F}_p$  上是可约的 (或不可约的)。

iii) 对于  $p=2, f(x) = x^2 - 2x + 2 = x^2 - 0x + 0 = x^2$ , 因此  $f(x)$  在  $\mathbb{F}_2$  上是可约的。

小于 10 且模 4 与 1 同余的仅有的奇素数是 5, 由于  $-1 \equiv 4 \equiv 2^2(\bmod 5)$ , 即  $\sqrt{-1} \equiv 2(\bmod 5)$ , 我们可以得到  $f(x)$  在  $\mathbb{F}_5$  上的完全分解式为

$$f(x) = (x - 1 - \sqrt{-1})(x - 1 + \sqrt{-1}) = (x - 1 - 2)(x - 1 + 2) = (x + 2)(x + 1)$$

$-1$  在  $\mathbb{F}_5$  上的另一平方根是 3。读者可验证, 根 3 给出的  $f(x)$  在  $\mathbb{F}_5$  上的分解式与根 2 给出的分解式是相同的。□

#### 5.4.2.2 利用不可约多项式构造域

让我们利用不可约多项式来构造有限域。

**定义 5.21 集合  $A[x]$  模一个多项式** 令  $A$  为一个代数结构,  $f, g, q, r \in A[x], g \neq 0$  满足除式 (5.4.4), 我们称  $r$  为  $f$  除以  $g$  的余式, 记为  $r \equiv f(\bmod g)$ 。

$A[x]$  中所有多项式模  $g$  的余式集称为  $A[x]$  模  $g$  中的多项式, 记为  $A[x]_g$ 。

类似于整数模一个正整数的情形,  $A[x]_f$  是所有次数小于  $\deg(f)$  的多项式的集合。

**定理 5.5** 令  $F$  是一个域,  $f$  是  $F[x]$  中的一个非零多项式。那么  $F[x]_f$  是一个环, 当且仅当  $f$  在  $F$  上不可约时,  $F[x]_f$  是一个域。

**证明** 首先, 在由式 (5.4.2)、式 (5.4.3) 和式 (5.4.4) 定义的模  $f$  加法和模  $f$  乘法运算下,  $F[x]_f$  显然是一个环, 它的零元和单位元与  $F$  的相同。

其次, 令  $F[x]_f$  是一个域。设  $f = gh$ , 其中  $g, h$  是  $F[x]$  中的非常数多项式。因为  $0 < \deg(g) < \deg(f), 0 < \deg(h) < \deg(f)$ , 所以  $g, h$  是  $F[x]_f$  中的非零多项式, 而  $f$  是  $F[x]_f$  中的零多项式。这破坏了  $F[x]_f$  中乘法群的封闭律, 因此  $F[x]_f$  不为域, 这与  $F[x]_f$  是域的假设相矛盾。

最后,令  $f$  在  $F$  上不可约,因为  $F[x]_f$  是一个环,所以我们只需证明  $F[x]_f$  中的非零元在  $F[x]_f$  中都有乘法逆。令  $r$  是  $F[x]_f$  中的一个非零多项式,  $\gcd(f, r) = c$ 。因为  $\deg(r) < \deg(f)$  且  $f$  不可约,所以  $c$  一定是一个常数多项式,可写做  $r = cs$ , 其中  $c \in F, s \in F[x]_f, \gcd(f, s) = 1$ 。类似于整数的情形,我们能够使用多项式的扩展欧几里得算法计算  $s^{-1}(\bmod f) \in F[x]_f$ , 而且由于  $c \in F$ , 存在  $c^{-1} \in F$ , 因此有  $r^{-1} = c^{-1}s^{-1} \in F[x]_f$ 。□

对有限域,不可约多项式  $f$  称为域的定义多项式。

**定理 5.6** 令  $F$  为含有  $p$  个元素的域,  $f$  是  $F$  上的  $n$  次不可约多项式,那么域  $F[x]_f$  中元素的个数是  $p^n$ 。

**证明** 由定义 5.21 我们知道  $F[x]_f$  是  $F[x]$  中所有次数小于  $\deg(f) = n$ 、系数取遍  $F$  中  $p$  个元的多项式全体构成的集合。在  $F[x]_f$  中恰有  $p^n$  个这样的多项式。□

**推论 5.4** 对于每一个素数  $p$  和每一个正整数  $n$ , 都存在一个含有  $p^n$  个元素的有限域。□

如推论 5.4 所指出的,由于  $F$  是素域  $\mathbb{F}_p$ , 域  $\mathbb{F}_p[x]_f$  的结构是很清楚的: 它仅是所有次数小于  $n$ 、系数在  $\mathbb{F}_p$  中的所有多项式的集合。在同构的作用下,我们甚至可以说  $\mathbb{F}_p[x]_f$  是阶为  $p^n$  的有限域。

**例 5.17 有限域元素的整数表示** 多项式  $f(x) = x^8 + x^4 + x^3 + x + 1$  在  $\mathbb{F}_2$  上不可约,  $\mathbb{F}_2$  上模  $f(x)$  的所有多项式集合构成一个含  $2^8$  个元素的域,域中元素是  $\mathbb{F}_2$  上所有次数小于 8 的多项式。因此域  $\mathbb{F}_2[x]_f$  中的元素有如下的形式:

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0$$

其中  $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0 \in \mathbb{F}_2$ 。因此该域中的任意元都可以表示成一个 8 比特  $b_7b_6b_5b_4b_3b_2b_1b_0$  或一字节的二进制整数。在十六进制编码中,我们可以用一个字符来作为由 4 比特表示的整数值的码字:

$$'0' = 0000 (=0), \dots, '9' = 1001 (=9), 'A' = 1010 (=10), \dots, 'F' = 1111 (=15)$$

由于一字节有 8 比特,所以一字节的十六进制编码能够用两个字符 'XY' 表示,其中 ' $0 \leq X \leq F, 0 \leq Y \leq F$ '. 也就是说,域  $\mathbb{F}_2[x]_f$  中的任一元都可以看做是在区间 ['00', 'FF'] 上的一个字节。

相反,在区间 ['00', 'FF'] 上的任一字节都可以看做是域  $\mathbb{F}_2[x]_f$  中的一个元素。例如,字节 01010111 (或十六进制值 '57') 对应的元素(多项式)是

$$x^6 + x^4 + x^2 + x + 1$$

□

由推论 5.4 和例 5.17, 我们可将域  $\mathbb{F}_2[x]_f$  看做是由最多  $\deg(f)$  个二进制比特的所有非负整数所构成的域。显然,这个域有  $2^{\deg(f)}$  个元。因此,对任意自然数  $n > 0$ , 集合  $\{0, 1\}^n$  构成一个含有  $2^n$  个元素的域,我们将这个域称为“ $n$  比特二进制域”。这个域中的运算类似于  $\mathbb{F}_2$  上次数小于  $n$  的多项式之间的运算。加法很简单,如例 5.18 所示。

**例 5.18** 令  $f$  是  $\mathbb{F}_2$  上 8 次的不可约多项式。在 8 比特二进制域中,加法类似于加系数模 2 的多项式加法(因此  $1 + 1 = 0$ )。例如(在十六进制中), ' $57$ ' + ' $83$ ' = ' $D4$ ':

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2$$

因此,这个域中的加法与定义多项式  $f$  无关。□

域  $\mathbb{F}_2[x]_f$  中的乘法与定义多项式  $f$  有关:它是两个模  $f$  多项式之间的乘法。利用多项式的扩展欧几里得算法可进行模运算。以后(在例 5.19 中)我们将给出实现域乘法的另一种方法,这种方法基于不同的域表示。

$n$  比特二元域由于可用整数简单自然地表示使它成为一个很有用的域,它在编码和密码学中都有很多应用。一种新的加密标准,高级加密标准(AES),就是在 8 比特二进制域上实现的。我们将在第 7 章中介绍 AES。

最后我们注意到,在定理 5.6 中我们从未假设  $p$  为素数。事实上,在定理 5.5 中,  $F$  可以是任意域,称  $F[x]_f$  为由基域  $F$  通过域扩张得到的扩域。既然  $F$  可以是任意域,它当然可以由另一基域得到的扩域。在许多有限域的应用中,我们需要知道更多的关于扩域和基域之间关系的信息(例如,在我们以后研究 AES 的时候,我们就需要知道这个关系)。使用有限域的不同表示方法也可以简化计算(例如,如果我们使用另一种不同的域表示方法,就可以不必使用欧几里得算法,减少例 5.18 中乘法的计算量)。下一节将使我们对有限域的结构有一个更好的理解。

### 5.4.3 用多项式基构造有限域

本节旨在提供一些知识来帮助我们更好地理解一些基于一般形式有限域的密码体制。我们在介绍时假定读者熟悉线性代数中的向量空间知识。但跳过本节不会对本书其余的绝大部分内容的阅读造成困难。

在 5.4.2 节中,我们给出了在同构意义下,域  $\mathbb{F}_p[x]_f$  就是阶为  $p^{\deg(f)}$  的有限域。然而,通常采用模一个不可约多项式的有限域对我们来说常常不是太方便的。在关于代数学基础的最后一部分内容里,我们用有限域  $F$  上不可约多项式的根来构造有限域。在实用中更经常采用这种方法构造域。

令  $F$  是有限域,  $n$  是任意正整数,  $f(x)$  是  $F$  上  $n$  次不可约多项式。由于在某处  $f(x)$  可以分解为  $n$  个线性多项式,因此我们知道  $f(x)$  在那里恰有  $n$  个根。我们马上就会看到“在某处”或“在那里”恰恰就是我们要构造的空间。

将  $f(x)=0$  的  $n$  个根记为

$$\theta_0, \theta_1, \dots, \theta_{n-1} \quad (5.4.6)$$

既然  $f(x)$  在  $F$  上不可约,因此这些根就都不在  $F$  中。

**定理 5.7** 令  $F$  是任意有限域,  $f(x) \in F[x]$  是  $F$  上  $n$  次不可约多项式,那么对于  $f(x)=0$  的任一根  $\theta$ , 元素

$$1, \theta, \theta^2, \dots, \theta^{n-1}$$

在  $F$  上是线性无关的,也就是说对于  $r_i \in F, i=0, 1, 2, \dots, n-1$  有:

$$r_0 + r_1 \theta + r_2 \theta^2 + \dots + r_{n-1} \theta^{n-1} = 0 \text{ 可推出 } r_0 = r_1 = \dots = r_{n-1} = 0 \quad (5.4.7)$$

**证明** 令  $\theta$  是  $f(x)=0$  的任一根,既然域  $F$  包含 1,而  $f(x)$  在域  $F$  上不可约,我们知道

$\theta \neq 1$ 。假定元素  $1, \theta, \theta^2, \dots, \theta^{n-1}$  在  $F$  上线性相关, 即存在  $r_i \in F, r_i$  不全为零 ( $i = 0, 1, 2, \dots, n-1$ ), 使得线性组合 (5.4.7) 式成立, 这等价于  $\theta$  是下式的一个根:

$$r(x) = r_0 + r_1 x + r_2 x^2 + \dots + r_{n-1} x^{n-1} = 0$$

其中  $r_i \in F (i = 0, 1, \dots, n-1)$ , 由定义 5.21,  $r(x)$  是域  $F[x]_f$  中的一个元, 因此  $r(x) = 0$  意味着  $r(x) \equiv 0 \pmod{f(x)}$ 。令  $a_n$  是  $f(x)$  的首系数, 那么  $a_n \in F, a_n \neq 0$  且  $a_n^{-1} f(x) \mid r(x)$ 。但由于  $a_n^{-1} f(x)$  是  $n$  次多项式, 而  $r(x)$  是小于  $n$  次的多项式, 所以这是不可能的, 除非  $r(x)$  是零多项式。这与  $r_i \in F, r_i$  不全为零 ( $i = 0, 1, \dots, n-1$ ) 的假定条件相矛盾。□

**定义 5.22 多项式基** 令  $F$  是有限域,  $f(x)$  是  $F$  上  $n$  次不可约多项式, 那么对于  $f(x) = 0$  的任一根  $\theta$ , 元素  $1, \theta, \theta^2, \dots, \theta^{n-1}$  称为  $F$  上的 (有限向量空间的) 一个 (多项式) 基。

由线性代数的知识, 我们知道  $n$  个元素的基可张成一个  $n$  维的向量空间, 这个扩张用  $F$  中的元素做标量, 即扩张成的空间具有下面的结构:

$$\left\{ \sum_{i=0}^{n-1} r_i \theta^i \mid r_0, r_1, \dots, r_{n-1} \in F \right\} \quad (5.4.8)$$

**定理 5.8** 令  $F$  是有限域,  $f(x)$  是  $F$  上  $n$  次不可约多项式, 那么对于  $f(x) = 0$  的任一根  $\theta$ , 式 (5.4.8) 中的向量空间是一个有  $(\#F)^n$  个元素的有限域。

**证明** 首先我们证明式 (5.4.8) 中的空间是一个环, 仅有的非平凡部分就是要证明对乘法满足封闭律。为了证明这一点, 我们注意到由

$$f(\theta) = a_n \theta^n + a_{n-1} \theta^{n-1} + \dots + a_0 = 0 \quad (5.4.9)$$

$a_n \in F, a_n \neq 0$ , 我们有

$$\theta^n = a_n^{-1} (-a_{n-1} \theta^{n-1} - \dots - a_0)$$

所以  $\theta^n$  是基  $1, \theta, \theta^2, \dots, \theta^{n-1}$  的一个线性组合。式 (5.4.9) 两边乘以  $\theta$ , 我们进一步可推导出, 对于任意正整数  $m \geq n$ ,  $\theta^m$  能够表示成同样基的线性组合。因此, 对于式 (5.4.8) 的空间中的任意  $u$  和  $v$ ,  $uv$  可表为  $1, \theta, \theta^2, \dots, \theta^m$  的一个线性组合, 其中  $m \leq 2(n-1)$ 。所以  $uv$  一定是基  $1, \theta, \theta^2, \dots, \theta^{n-1}$  的一个线性组合, 因此它在式 (5.4.8) 的空间里, 这样我们就证明了封闭律。

其次, 为了证明式 (5.4.8) 中的空间是一个域, 我们只需证明这个空间不包含零因子。为了证明这一点, 我们可以利用式 (5.4.7) 中线性无关的关系来验证对于  $uv = 0$ , 或者  $u$  的系数或者  $v$  的系数一定全为零, 因此  $u = 0$  或  $v = 0$ 。

最后, 注意到由于扩张过程中使用了  $F$  的  $\#F$  个元素作为标量并使用了  $n$  个元素的基, 因此该扩张空间恰有  $(\#F)^n$  个元素。□

**定义 5.23 有限域  $\mathbb{F}_{q^n}$**  令  $q$  是有限域  $F$  中的元素个数, 由  $n$  个元素组成的基在  $F$  上的扩张所构成的有限域记为  $\mathbb{F}_{q^n}$ 。

**定理 5.9** 令  $F$  是  $q$  个元素的有限域,  $\mathbb{F}_{q^n}$  是  $F$  上张成的有限域, 那么

- i)  $\mathbb{F}_{q^n}$  的特征与  $F$  的特征相同;
- ii)  $F$  是  $\mathbb{F}_{q^n}$  的一个子域;

iii) 对任一元素  $a \in \mathbb{F}_{q^n}$ ,  $a^q = a$  当且仅当  $a \in F$ 。

证明 令  $1, \theta, \theta^2, \dots, \theta^{n-1}$  是  $\mathbb{F}_{q^n}$  在  $F$  上的一组基。

i)  $F$  的特征记为  $\text{char}(F)$ , 那么将  $\mathbb{F}_{q^n}$  上的任一元自加  $\text{char}(F)$  次, 我们得到

$$\sum_{i=0}^{n-1} \text{char}(F) r_i \theta^i = \sum_{i=0}^{n-1} 0 \theta^i = 0$$

因此  $\text{char}(\mathbb{F}_{q^n}) = \text{char}(F)$ 。

ii) 由于基包含 1, 并用  $F$  中元素做标量, 则  $F$  中任意元都是 1 的线性组合, 因此是基的线性组合。

iii) ( $\Leftarrow$ ) 考虑子域  $F = \{0\} \cup F^*$ , 其中  $F^*$  是非零元的乘法群, 所以对任意  $a \in F$ , 或者  $a = 0$ , 或者  $a \in F^*$ 。前一种情况显然满足  $a^q = a$ 。对于后一种情况, 由拉格朗日定理(推论 5.2),  $\text{ord}(a) \mid \# F^* = q - 1$ , 所以  $a^{q-1} = 1$ 。因此  $a^q = a$  也成立。

( $\Rightarrow$ ) 满足  $a^q = a$  的任意  $a \in \mathbb{F}_{q^n}$  一定是多项式  $x^q - x = 0$  的根, 这个多项式的次数为  $q$ , 因此在  $\mathbb{F}_{q^n}$  中包括 0 在内最多有  $q$  个根。由 ii),  $F$  是  $\mathbb{F}_{q^n}$  的子域, 它已经包含了  $x^q - x = 0$  的所有根, 所以  $\mathbb{F}_{q^n}$  中的其他元都不会是  $x^q - x$  的根。□

在我们讨论由  $q$  元域  $F$  扩张成域  $\mathbb{F}_{q^n}$  时, 我们从未假定或者要求  $q$  是一个素数, 也就是说, 我们从未假定或是要求  $F$  是一个素域。下面的定理给出了  $F$  和  $F$  的扩域  $\mathbb{F}_{q^n}$  之间的关系, 并且指明了  $q$  的性质。

**定理 5.10 子域判定准则** 令  $p$  是一个素数, 那么  $F$  是  $\mathbb{F}_{p^n}$  的子域, 当且仅当  $F$  有  $p^m$  个元素, 其中  $m$  是  $n$  的一个正因子。

证明 ( $\Rightarrow$ ) 令  $F$  是  $\mathbb{F}_{p^n}$  的子域,  $F = \mathbb{F}_p$  或  $F = \mathbb{F}_{p^n}$  是两种平凡的情形。令  $F$  是  $\mathbb{F}_{p^n}$  的不同于  $\mathbb{F}_p$  的真子域。由定理 5.9(i),  $\mathbb{F}_{p^n}$  有特征  $p$ , 因此  $F$  也有特征  $p$ 。所以  $F$  包含  $\mathbb{F}_p$  作为子域, 对于某个  $m, 1 \leq m \leq n$ ,  $F$  由  $m$  个元素组成的一组基在  $\mathbb{F}_p$  上扩张而成。我们只需要证明  $m \mid n$ 。两个乘法群  $\mathbb{F}_{p^n}^*$  和  $F^*$  分别有  $p^n - 1$  和  $p^m - 1$  个元, 既然后者是前者的子群, 由拉格朗日定理(定理 5.1),  $p^m - 1 \mid p^n - 1$ 。只有当  $m \mid n$  时, 才有  $p^m - 1 \mid p^n - 1$ 。

( $\Leftarrow$ ) 令  $m$  是  $n$  的一个正因子,  $F$  是有  $p^m$  个元素的域, 由于  $n/m$  是一个正整数, 利用  $F$  上的一个  $n/m$  次不可约多项式, 我们能够扩张成一个  $(p^m)^{n/m} = p^n$  个元素的域。将这个扩域记为  $\mathbb{F}_{p^n}$ , 由定理 5.9(ii),  $F$  是  $\mathbb{F}_{p^n}$  的子域。□

令  $f(x)$  是  $\mathbb{F}_p$  上任意  $n$  次不可约多项式。由定理 5.6, 我们现在知道  $\mathbb{F}_{p^n}$  同构于  $\mathbb{F}_p[x]_f$ 。尽管两个同构的域应该视为没有什么不同, 但使用一个域也可能会比使用另一个域更容易。的确, 对于  $\mathbb{F}_{p^n}$ , 子域判定准则定理更容易的证明就给出了这样一个清楚的证据。下面的例子给出了另一个证据。

**例 5.19 域  $\mathbb{F}_{2^8}$**  我们已经看到  $\mathbb{F}_2[x]_{x^8+x^4+x^3+x+1}$  (在例 5.18 中) 是  $\mathbb{F}_2$  上以不可约多项式  $x^8+x^4+x^3+x+1$  取模的所有多项式构成的集合, 它包含  $2^8$  个元素。现在我们知道  $\mathbb{F}_{2^8}$  也是有  $2^8$  个元素的域, 可以用下面的空间来表示

$$\{b_7\theta^7 + b_6\theta^6 + b_5\theta^5 + b_4\theta^4 + b_3\theta^3 + b_2\theta^2 + b_1\theta + b_0\}$$



其中,  $\theta$  是(例如)方程  $x^8 + x^4 + x^3 + x + 1 = 0$  的一个根, 标量  $b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0 \in \mathbb{F}_2$ 。显然, 这两个域是同构的; 特别地, 在  $\mathbb{F}_{2^8}$  的后一种表示中也可以用一个字节来表示一个元素。

在例 5.18 中我们提到在  $\mathbb{F}_2[x]_{x^8+x^4+x^3+x+1}$  中的乘法有点复杂, 需要模一个多项式, 在多项式除法中要用欧几里得算法。由多项式基扩张成的  $\mathbb{F}_{2^8}$  中的乘法可以更容易: 直接乘两个元素, 用基  $1, \theta, \dots, \theta^7$  的线性组合来表示结果中任一个带有  $\theta^i, i > 7$  的项。

例如, 让我们计算 '57' · '83', 或者

$$(\theta^6 + \theta^4 + \theta^2 + \theta + 1) \cdot (\theta^7 + \theta + 1) = \theta^{13} + \theta^{11} + \theta^9 + \theta^8 + \theta^6 + \theta^5 + \theta^4 + \theta^3 + 1$$

由于

$$\theta^8 + \theta^4 + \theta^3 + \theta + 1 = 0$$

我们有下面的线性组合(注意在  $\mathbb{F}_2$  中  $-1 = 1$ ):

$$\theta^8 = \theta^4 + \theta^3 + \theta + 1$$

$$\theta^9 = \theta^5 + \theta^4 + \theta^2 + \theta$$

$$\theta^{11} = \theta^7 + \theta^6 + \theta^4 + \theta^3$$

$$\theta^{13} = \theta^9 + \theta^8 + \theta^6 + \theta^5$$

因此,

$$\theta^{13} + \theta^{11} + \theta^9 + \theta^8 + \theta^6 + \theta^5 + \theta^4 + \theta^3 + 1 = \theta^7 + \theta^6 + 1$$

即我们有 '57' · '83' = 'C1'。□

我们现在给出一个注释作为对有限域研究的小结。

**注释 5.2** 我们已经讨论了构造有限域的两种方法: 模一个不可约多项式得到的域(5.4.2 节)和由一个多项式基扩张成的域(5.4.3 节)。在我们对有限域的讨论中, 我们用  $\mathbb{F}_q$  来表示后一种构造的域。然而, 在同构意义下, 元素数目相同的两个域可以看做是一样的。因此从现在开始, 我们将任意  $q$  个元素的有限域记做  $\mathbb{F}_q$ , 这里  $q$  是一个素数幂。□

#### 5.4.4 本原根

在 4.4.4.1 节<sup>①</sup>中, 我们断言  $n-1$  的完全分解为使用有效的确定算法来回答  $n$  是否为素数提供了一条“内在的信息”(也就是验证一个  $\mathcal{NP}$  问题的辅助性输入)。现在用有限域的知识, 我们能够很容易地证明这个断言。

**定理 5.11** 域的乘法群是循环群。

**证明** 由定理 5.9(iii), 多项式  $x^{p^n-1} - 1 = 0$  的所有根构成了  $(\mathbb{F}_{p^n})^*$ 。然而, 这个多项式的所有根是 1 的  $p^n - 1$  个不同的(非平凡的)根, 它们分布在一个单位圆上。所以存在 1 的一个  $p^n - 1$  次根, 它生成群  $(\mathbb{F}_{p^n})^*$ 。因此  $(\mathbb{F}_{p^n})^*$  是循环群。□

**定义 5.24 本原根** 群  $(\mathbb{F}_{p^n})^*$  的乘法生成元称为域  $\mathbb{F}_{p^n}$  的本原根。

① 原文为 4.5, 可能有误。

**定理 5.12** 令  $n$  是一个正整数,  $n-1=r_1 r_2 \cdots r_k$  是  $n-1$  的完全素分解(素因子可重复), 那么  $n$  是素数, 当且仅当存在一个正整数  $a < n$ , 满足  $a^{n-1} \equiv 1 \pmod{n}$  且  $a^{(n-1)/r_i} \not\equiv 1 \pmod{n}$ ,  $i=1, 2, \cdots, k$ 。

**证明** ( $\Rightarrow$ ) 如果  $n$  是一个素数, 那么由定理 5.11, 群  $(\mathbb{F}_p)^*$  是循环群且有一个生成元, 它是 1 的  $n-1$  次根。将这个根记为  $a$ , 那么  $a$  满足定理陈述中的条件。

( $\Leftarrow$ ) 设整数  $a < n$  满足定理陈述中的条件, 那么  $a, a^2, \cdots, a^{n-1}$  是  $x^{n-1} - 1 \equiv 0 \pmod{n}$  的解。对于任意的  $1 \leq i < j \leq n-1$ , 一定有  $a^i \not\equiv a^j \pmod{n}$ , 否则假定  $a^{j-i} \equiv 1 \pmod{n}$ , 对于某个  $i, j, 0 < j-i < n-1$ , 那么由定义 5.9,  $\text{ord}(a) \mid j-i \mid n-1$ , 这与定理陈述中的条件相矛盾。现在我们知道  $\langle a \rangle$  是  $n-1$  个元素的乘法群(模  $n$  乘法), 这个群最多能包含  $\phi(n)$  个元素。所以  $\phi(n) = n-1$ 。因此由欧拉函数的定义(定义 5.11),  $n$  是一个素数。  $\square$

定理 5.12 给出了找模素数  $p$  的一个本原根, 也就是找群  $\mathbb{F}_p^*$  的生成元的一个有效算法, 该算法在算法 5.1 中列出。

#### 算法 5.1 模一个素数的任一本原根

输入  $p$ : 一个素数;  $q_1, q_2, \cdots, q_k$ :  $p-1$  的所有素因子;

输出  $g$ : 模  $p$  的任一本原根。

PrimitiveRoot( $p, q_1, q_2, \cdots, q_k$ )

1. 取  $g \in_v [2, p-1]$ ;
2. for( $i=1, i++, k$ ) do  
    如果  $(g^{(p-1)/q_i} \equiv 1 \pmod{p})$ , (返回 PrimitiveRoot( $p, q_1, q_2, \cdots, q_k$ ));
3. 返回( $g$ )。

由定理 5.2(4), 我们知道在群  $\mathbb{F}_p^*$  中恰有  $\phi(p-1)$  个阶为  $p-1$  的元, 这些元素都是这个群的生成元, 因此算法 5.1 将在

$$\frac{p-1}{\phi(p-1)} < 6 \log \log p - 1$$

步递归调用后结束(见[200]中的 65 页)。因为  $p-1$  的素因子个数上限为  $\log p$ , 所以这个算法的时间复杂度上限为  $O_B((\log p)^4 \log \log p)$ 。

## 5.5 用椭圆曲线上的点构造群

现代密码学中非常重要的一类群就是由椭圆曲线上的点构造的群。Miller[205]和 Koblitz[168]最先提出用椭圆曲线群来实现公钥密码学。

密码学中使用的椭圆曲线定义在像有限域这样的有限代数结构上。为便于说明, 我们只讨论特征大于 3 的素域  $\mathbb{F}_p$  这种比较简单的情形。这样的曲线就是下面形式的方程的几何解  $P = (x, y)$  所构成的集合

$$E: y^2 = x^3 + ax + b \pmod{p} \quad (5.5.1)$$

其中  $a$  和  $b$  是满足  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ <sup>①</sup> 的  $\mathbb{F}_p$  ( $p > 3$ ) 中的常数。为使  $E$  上的点构成一个群, 还需要另外包含一个记为  $\mathcal{O}$  的点。称这个额外的点为无穷远点, 表示为

$$\mathcal{O} = (x, \infty)$$

因此对于这个群的形式, 我们可写为

$$E = \{P = (x, y) \mid \text{由式(5.5.1)解出的 } x, y \in \mathbb{F}_p\} \cup \{\mathcal{O}\} \quad (5.5.2)$$

这个点集在群运算下构成群, 习惯上群运算写做加法, 记为“+”。我们很快就要定义这个运算。

将式(5.5.1)右边的三次多项式记为  $f(x)$ , 如果  $f(x)$  在  $\mathbb{F}_p$  上可约, 那么对于  $f(x)$  的零点  $\xi \in \mathbb{F}_p$  [也就是  $f(\xi) \equiv 0 \pmod{p}$ ], 点  $(\xi, 0) \in E$ 。我们很快将看到在群运算“+”下, 这些点的阶为 2。既然  $f(x)$  是三次多项式, 所以最多有三个这样的点 [按照  $f(x)$  在  $\mathbb{F}_p$  上可约性的不同情况, 有 1 个点或 3 个点; 通过习题 5.13 找出理由]。

所有不为  $\mathcal{O}$  的点都可以通过计算  $\eta \in \mathbb{F}_p$ , 使得  $f(\eta) \not\equiv 0 \pmod{p}$  是  $\mathbb{F}_p$  中的二次剩余得到 (也就是模  $p$  的一个平方数, 见 6.5 节)。在上述情形中, 对于每一个这样的  $\eta$ ,  $y$  都有两个不同的解 (在  $\mathbb{F}_p$  中每一个二次剩余项都有两个模  $p$  平方根, 见推论 6.2)。既然  $f(\eta)$  是常数, 两个平方根就是  $\sqrt{f(\eta)}$  和  $-\sqrt{f(\eta)}$ 。因此, 我们可以将两个解对应的点记为  $(\eta, \sqrt{f(\eta)})$  和  $(\eta, -\sqrt{f(\eta)})$ 。

最后我们知道, 对于  $\mathbb{F}_p$  中的所有元  $\xi, \eta$ , 满足  $f(\xi) \equiv 0 \pmod{p}$ ,  $f(\eta)$  为  $\mathbb{F}_p$  中的二次剩余, 曲线  $E(\mathbb{F}_p)$  上的点是  $\mathcal{O}, (\xi, 0), (\eta, \sqrt{f(\eta)})$  和  $(\eta, -\sqrt{f(\eta)})$ 。

### 5.5.1 群运算

式(5.5.2)中定义的集合  $E$  在下面定义的运算“+”下构成一个阿贝尔群。

**定义 5.25 椭圆曲线群运算 (“弦切法”)** 令  $P, Q \in E$ ,  $\ell$  是通过  $P$  和  $Q$  的直线 (如果  $P = Q$ ,  $\ell$  就是与  $E$  相切的切线),  $R$  是  $\ell$  与  $E$  相交的第三个点。令  $\ell'$  是  $R$  与  $\mathcal{O}$  的连线, 那么  $P$  “+”  $Q$  就是  $\ell'$  与  $E$  相交的第三个交点, 即  $\ell'$  与  $E$  相交于  $R, \mathcal{O}$  和  $P$  “+”  $Q$ 。

我们暂时假定, 在定义 5.25 下,  $(E, “+”)$  确实构成了一个群。我们首先解释为什么要求式(5.5.1)中三次多项式的系数满足  $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$ 。注意到

$$\Delta = -4a^3 - 27b^2$$

是三次多项式  $f(x) = x^3 + ax + b$  的判别式。如果  $\Delta \equiv 0 \pmod{p}$ , 那么  $f(x) \equiv 0 \pmod{p}$  至少有一个二重零点  $X$  [使  $f(X) \equiv 0$  的根], 显然  $(X, 0)$  在  $E$  上。对于  $E(x, y) = y^2 - x^3 - ax - b = 0$ , 这个点满足

$$\frac{\partial E}{\partial y} = 2y \Big|_{y=0} = \frac{\partial E}{\partial x} \Big|_{x=X} = 0$$

即  $(X, 0)$  是一个奇异点, 在这一点处的切值没有定义。所以在奇异点  $(X, 0)$  上, 弦切运算不能成立,  $E$  不构成群。

① 理由将在定义 5.25 后面给出。

图 5.1 给出了弦切运算的图示,上面的曲线是  $\Delta < 0$  的情形(三次多项式只有一个实根),下面的曲线是  $\Delta > 0$  的情形。我们用点来绘制曲线是为了表明  $E(\mathbb{F}_p)$  是一个离散集合,这些离散点称为  $\mathbb{F}_p$ -有理点,它们的个数是有限的[见即将给出的式(5.5.6)]。

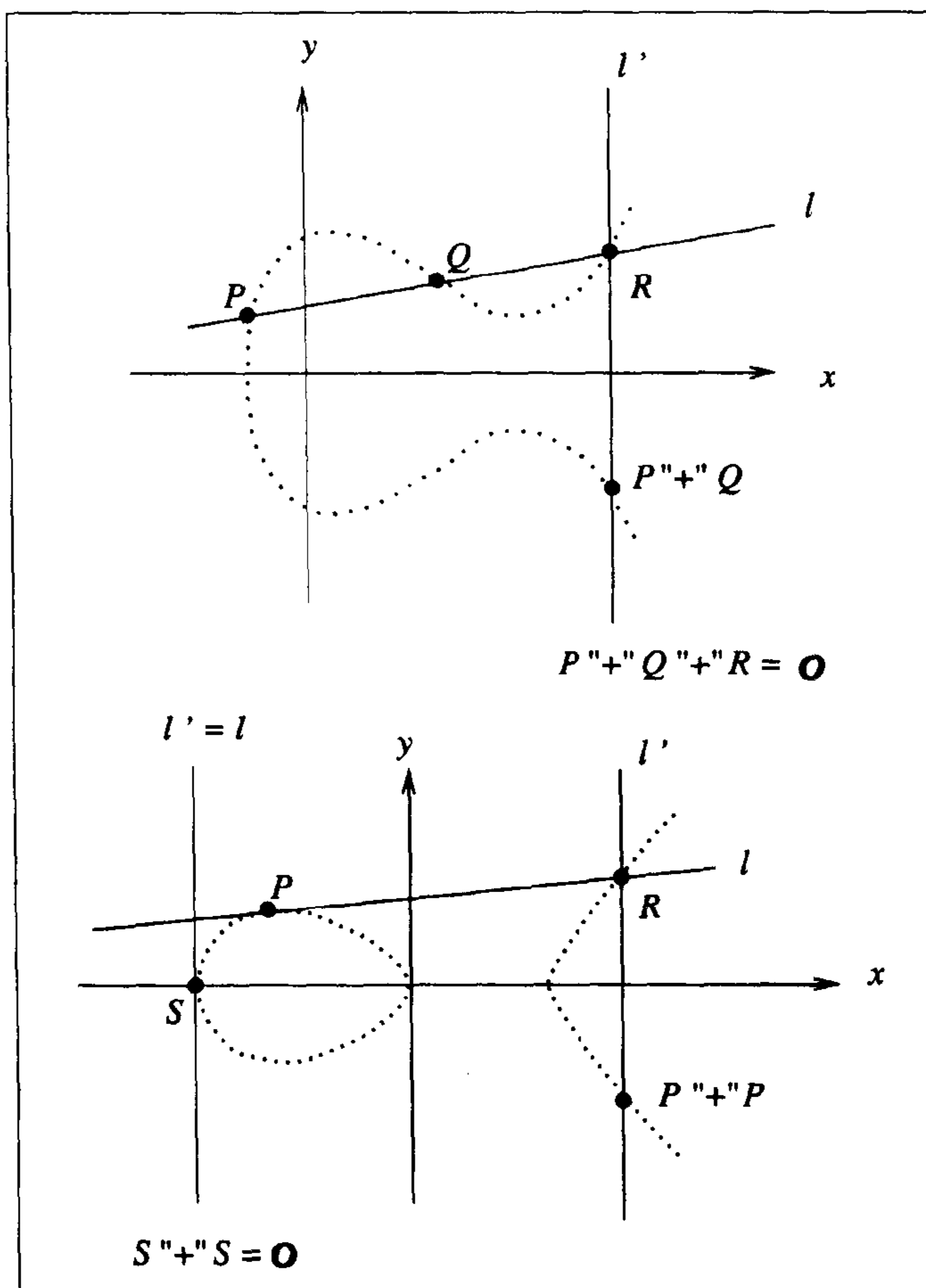


图 5.1 椭圆曲线群运算

现在让我们证明在弦切运算下,  $(E, "+")$  的确构成一个群。

首先,对任意的  $P = (X, Y) \in E$ ,对于  $Q$  是  $O$  的特殊情形应用定义 5.25(既然  $E$  中包含  $O$ )。通过  $P$  和  $O$  的直线  $l$  是

$$l: x = X$$

由  $P \in E$  可得,  $f(X) = X^3 + aX + b \pmod{p}$  为  $\mathbb{F}_p$  中的二次剩余,因此  $-Y$  是  $y^2 = f(X)$  的另一个解[也就是  $f(X)$  模  $p$  的另一个平方根]。也就是说,  $l$  也交于另一点  $R = (X, -Y) \in E$ 。显然,因为  $l' = l$ ,所以  $l'$  与  $l$  交于同样的三个点。由定义 5.25,对任意  $P \in E$ ,我们有

$$P = P'' + O$$

而且,对所有  $(x, y) \in E$ ,我们也推导出了

$$(x, y) + (x, -y) = O$$

记  $(x, -y) = "-"(x, y)$ , 我们看出点  $\mathcal{O}$  恰恰就是在运算 "+" 下的单位元。因此, 我们得到了  $(E, "+")$  的单位元律和可逆律。

这种特殊情形的一个特例就是  $y_1 = -y_2 = 0$ 。这就是  $P = "-"P$  的情形(在图 5.1 中下面曲线上的点  $S$ )。对于这个二重特殊点, 我们有  $P "+" P = \mathcal{O}$ 。也就是说,  $P$  是一个二阶元。我们前面提到过这个特殊的元素: 它就是  $y^2 = f(x) \equiv 0 \pmod{p}$  的解。只有当  $f(x)$  在  $\mathbb{F}_p$  中有零点时, 也就是当  $f(x)$  在  $\mathbb{F}_p$  上可约时, 这样特殊的点才存在。

现在让我们考虑  $\ell$  不是垂直直线的一般情形, 此时  $\ell$  的公式表达是

$$\ell: y - y_1 = \lambda(x - x_1) \quad (5.5.3)$$

其中

$$\lambda = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2} & \text{当 } x_1 \neq x_2 \text{ 时} \\ \frac{3x_1^2 + a}{2y_1} & \text{当 } x_1 = x_2 \text{ 且 } y_1 = y_2 \neq 0 \text{ 时} \end{cases} \quad (5.5.4)$$

因为  $\ell$  将与曲线交于  $R = (x_3, y_3)$ , 我们可以用式(5.5.1)和式(5.5.3)来寻找点  $R$ 。点  $R$  的  $x$  坐标就是下式的解

$$\ell \cap E: [\lambda(x - x_1) + y_1]^2 - (x^3 + ax + b) = 0$$

注意到  $\ell \cap E$  是一个三次多项式, 它的解为  $x_1, x_2$  和  $x_3$ , 我们也可以将它写为

$$\ell \cap E: c(x - x_1)(x - x_2)(x - x_3) = 0$$

这里  $c$  是某个常数。比较  $\ell \cap E$  的两种写法的系数( $x^3$  和  $x^2$  的系数), 我们得到  $c = -1$  且

$$x_3 = \lambda^2 - x_1 - x_2$$

最后由定义 5.25 和  $P "+" Q = "-"R$ , 我们得到点  $P "+" Q$  的坐标为

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1 \end{aligned} \quad (5.5.5)$$

这里,  $\lambda$  如式(5.5.4)中所定义。注意到因为直线  $\ell$  交于  $P, Q$  和  $R$ , 所以  $R$  一定在曲线上。因此,  $P "+" Q = "-"R$  也一定在曲线上。于是, 我们就获得了  $(E, "+")$  的封闭律。

一步步地应用式(5.5.5)可以验证结合律。由于所占篇幅较长而且方法单一, 我们就不在这里验证了, 而将它作为一个习题留给读者。

最后, 我们知道  $(E, "+")$  的确是一个群, 而且显然它是一个阿贝尔群。

**例 5.20** 因为  $4 \times 6^3 + 27 \times 4^2 \equiv 1 \not\equiv 0 \pmod{7}$ , 因此  $\mathbb{F}_7$  上的方程  $E: y^2 = x^3 + 6x + 4$  定义了一个椭圆曲线群。  $E(\mathbb{F}_7)$  上的点如下:

$$E(\mathbb{F}_7) = \{\mathcal{O}, (0, 2), (0, 5), (1, 2), (1, 5), (3, 0), (4, 1), (4, 6), (6, 2), (6, 5)\}$$

应用加法法则有  $(3, 0) "+" (3, 0) = \mathcal{O}$ ,  $(3, 0) "+" (4, 1) = (1, 2)$  和  $(1, 2) "+" (1, 2) = (0, 2)$ 。

读者可以验证, 例如,  $(1, 2)$  是这个群的生成元, 因此  $E(\mathbb{F}_7)$  是循环群。  $\square$

我们已经介绍了定义在素域  $\mathbb{F}_p$  ( $p > 3$ ) 上  $E$  的最简单情形的椭圆曲线群。通常情况下,  $E$  可以定义在  $\mathbb{F}_q$  上, 这里  $q$  是一个素数幂。对于  $p = 2$  和  $3$  的情形有点复杂, 但运算原理是相同的。对有兴趣的读者, 我们推荐[274]来进行进一步的研究。

### 5.5.2 点乘

从现在开始,我们将省略运算符号“+”和“-”中的引号。

令  $m$  是一个整数且  $P \in E$ , 记

$$[m]P = \begin{cases} \underbrace{P + P + \cdots + P}_m & \text{当 } m > 0 \text{ 时} \\ \mathcal{O} & \text{当 } m = 0 \text{ 时} \\ [-m] - P & \text{当 } m < 0 \text{ 时} \end{cases}$$

$[m]P$  的计算(或任意加法群中群运算的重数)类似于乘法群中的求幂,在算法 5.2 中给出。

---

#### 算法 5.2 椭圆曲线上元素的点乘

输入 点  $P \in E$ ; 整数  $m > 0$ ;

输出  $[m]P$ 。

EC\_Multiply( $P, m$ )

1. 如果  $m = 0$ , 则返回( $\mathcal{O}$ );
  2. 如果  $m \bmod 2 = 0$ , 则返回( $\text{EC\_Multiply}(P + P, m \div 2)$ );  
( $\ast \div$  表示整数中的除法, 也就是  $m \div 2 = \lfloor m/2 \rfloor \ast$ )
  3. 返回( $P + \text{EC\_Multiply}(P + P, m \div 2)$ )。
- 

例如, 执行  $\text{EC\_Multiply}(P, 14)$ , 算法 5.2 将进行下面四次递归调用:

$$\begin{aligned} \text{EC\_Multiply}(P, 14) &= \text{EC\_exp}(P + P, 7) && \text{(执行 2)} \\ &= [2]P + \text{EC\_Multiply}([2]P + [2]P, 3) && \text{(执行 3)} \\ &= [2]P + [4]P + \text{EC\_Multiply}([4]P + [4]P, 1) && \text{(执行 3)} \\ &= [2]P + [4]P + [8]P + \text{EC\_Multiply}([8]P + [8]P, 0) && \text{(执行 3)} \\ &= [2]P + [4]P + [8]P + \mathcal{O} && \text{(执行 1)} \end{aligned}$$

结果是  $[14]P$ 。

考虑到  $m \approx p$ , 以及在式(5.5.4)和式(5.5.5)中包含与  $p$  大小相仿的数的平方运算, 算法 5.12 的时间复杂度是  $O_B((\log p)^3)$ 。我们注意到算法 5.2 没有用到基域的任何性质, 因此它并不是一个有效的实现, 它的目的仅仅是提供一个如何计算点乘的简洁说明。几种实现计算点乘的更有效方法, 如预计算和利用特殊域的性质, 可以在[36]中找到。

### 5.5.3 椭圆曲线离散对数问题

与乘法的效果相逆, 即给定点对  $(P, [m]P)$ , 求整数  $m$ , 是一个与点乘区别非常大的问题。这个问题称为椭圆曲线离散对数问题或简称为 ECDLP。当点  $P$  有大的素数阶时, 普遍认为 ECDLP 是难解的(计算上不可行的)。

一般而言, 群中任意点的阶与该群的阶比例相关。据粗略估计, 域  $\mathbb{F}_q$  ( $q$  是一个素数幂) 上定义的一条曲线应粗略地有  $q$  个点, 因为  $x^3 + ax + b \in \mathbb{F}_q$  的  $q/2$  情形在  $\mathbb{F}_q$  中产生二次剩余



(如果  $x^3 + ax + b$  是  $\mathbb{F}_q$  上的一个排列, 则精确值  $(q-1)/2$  是偶阶群  $\mathbb{F}_q^*$  中的二次剩余), 且每个二次剩余情形可求解  $\mathbb{F}_q$  中的两个  $y$  值。Hasse 定理陈述为

$$\#E(\mathbb{F}_q) = q + 1 - t, \text{ 其中 } -2\sqrt{q} \leq t \leq 2\sqrt{q} \quad (5.5.6)$$

这里的  $t$  称为在  $q$  的“Frobenious 迹”。从这里我们可以看出,  $\#E(\mathbb{F}_q)$  与  $q$  是同量级的。对于定义在  $\mathbb{F}_q$  上的曲线(一般情形), 找一个略小于  $q$  的大素数  $p$ , 使得  $E(\mathbb{F}_q)$  包含一个  $p$  阶子群是非常容易的。求解 ECDLP 最好的已知算法的时间复杂度为  $O(\sqrt{r}) \approx O(\sqrt{q})$  (因为  $|r| \approx |q|$ )。这近乎利用生日悖论所进行的强力搜索法得到的结果。对于与  $q$  同量级的任意阿贝尔群上的离散对数问题, 都可以得到这样的结果。事实上, Pollard 的  $\lambda$  方法能够容易地改进成适用于 ECDLP 的情况。因此, 我们可以说, 解 ECDLP 的复杂度  $O(\sqrt{q})$  与所讨论的群结构无关。

对于有限域中的离散对数(将在定义 8.2 中正式定义), 存在称为指数积分的算法。求解有限域  $\mathbb{F}_q$  中离散对数的指数积分算法的时间复杂度有一个亚指数表达式  $\text{sub\_exp}(q)$ , 将在式(8.4.2)中给出。

复杂度表达式  $O(\sqrt{q})$  与  $q$  的规模成指数关系。对于同样的输入, 作为一个大数的函数  $O(\sqrt{q})$  要比亚指数函数  $\text{sub\_exp}(q)$  增长快得多。这意味着求解这两个问题, 在花费同样多时间的情况下, ECDLP 的基域要比普通离散对数问题所基于的有限域小得多。对于 ECDLP, 通常令  $q \approx 2^{160}$ 。此时抗强力搜索法的难度是  $2^{80}$  数量级的。为使有限域上的离散对数问题获得相似的难度, 亚指数表达式(8.4.2)就需要  $q$  达到  $2^{1000}$  数量级。我们进一步注意到, 随着硬件计算技术的发展,  $q$  也该相应地增大。由于  $\sqrt{q}$  和  $\text{sub\_exp}(q)$  的渐进差别非常大, 椭圆曲线情况中  $q$  的增长速度要比有限域情况中  $q$  的增长速度慢得多。

在一个相对较小的有限域上, ECDLP 的计算不可行性意味着椭圆曲线群在更有效的公钥密码体制的实现方面有很好的应用。既然公钥密码学也称为非对称密码学, 意思是用公钥加密是容易的, 而没有正确私钥的解密是困难的。因此, 我们说基于椭圆曲线群的公钥密码学要比基于有限域的公钥密码学更加非对称。

然而我们要先提醒大家, 有一些较弱的椭圆曲线, 对于这些较弱的情况,  $2^{160}$  数量级的有限域太小了。我们将在第 13 章看到这种较弱的情况和它那令人惊讶的实在应用。

## 5.6 本章小结

本章中, 在我们研究了抽象代数结构之后, 现在知道了例如群、环和域这样的代数结构都有算术运算的有限形式。例如, 我们已经看到了对于任意正整数  $n$ , 最多  $n$  个二进制比特的所有非负整数构成一个  $2^n$  个元素的有限域, 也就是说, 这个结构在加法和乘法下封闭(由于这些运算都是由最基本的加法和乘法运算导出的, 因此也在减法、除法和所有其他的诸如求幂、求根等代数运算下封闭)。有限空间中具有封闭性的代数结构为构造密码算法和协议提供了基石。

我们的内容不仅为大多数读者提供了一个完整的参考, 而且由于带有大量的理解性和说明性的材料, 所以有数学功底的读者也能够对这些主题有一个深入的理解。对抽象代数各主题的一个更全面的探讨可以在[179]中找到, 而关于椭圆曲线的研究则可以在如[274]中找到。

## 习题

- 5.1 在例 5.2(5)中,我们已经证明了  $\text{Fermat}(n)$  是  $\mathbb{Z}_n^*$  的子群。证明对于  $n$  为任意奇合数,  $\#\text{Fermat}(n) < \#\mathbb{Z}_n^*/2$ 。论述这个不等式是概率素性检测算法 4.5 实现原理的基础。
- 5.2 证明  $\text{DIV}3 = \{0\} \cup 3\mathbb{N}$  (集合  $\text{DIV}3$  在 4.3 节例 4.1 中定义)。
- 5.3 在群  $\mathbb{Z}_{11}^*$  中:(i)有多少生成元?(ii)找出它的所有生成元。(iii)找出它的所有子群。
- 5.4 令  $n$  是一个奇合数且不是素数的幂,群  $\mathbb{Z}_n^*$  有生成元吗?
- 5.5 运用例 5.10 给出的“齿轮标记”法证实群  $\mathbb{Z}_{35}^*$  中元素的最大阶是 12,并且任意元素的阶一定整除 12。
- 5.6 令  $n = pq$ ,其中  $p$  和  $q$  是不同的奇素数,证明前面问题的一般情形,即:(i) $\mathbb{Z}_n^*$  中元素的最大阶是  $\lambda(n) = \text{lcm}(p-1, q-1)$ ;(ii) $\mathbb{Z}_n^*$  中每一元素的阶均整除  $\lambda(n)$ 。
- 5.7 为什么有限环或域<sup>①</sup>的特征一定是素数?
- 5.8 利用多项式的长除法作为子程序,构造多项式的扩展欧几里得算法。
- 5.9 令  $n$  是任意自然数,构造  $n$  比特整数  $\{0,1\}^n$  的有限域。  
提示:利用例 5.17 中给出的映射法构造  $\mathbb{F}_2[x]_f$  与  $\{0,1\}^n$  之间的映射,这里  $f$  是  $\mathbb{F}_2$  上的  $n$  次多项式。
- 5.10  $\mathbb{F}_{2^8}$  有多少同构的子域?  $\mathbb{F}_{2^8}$  是其中之一吗?
- 5.11 为什么群的生成元也称为本原根?
- 5.12 对于奇数  $p$ ,知道  $p-1$  的完全分解,构造一个有效的算法来回答“ $p$  是素数吗?”这一问题,该算法的正确概率是 1[不使用  $\text{Prime\_Test}(p)$ ,由于它不能获得正确概率 1;也不使用试除法,因为它效率太低]。
- 5.13 对于  $\mathbb{F}_p$  上的椭圆曲线  $E: y^2 = x^3 + ax + b, p > 3$ ,证明如果  $f(x) = x^3 + ax + b$  在  $\mathbb{F}_p$  上不可约,那么  $E$  没有二阶元,否则,有 1 或 3 个二阶元。
- 5.14 对于 5.5.1 节中定义的群  $(E, "+")$ ,证明结合律成立。
- 5.15 证明例 5.20 中的点  $(1,2)$  是一个群生成元。

① 此处可能有问题,“有限环或域”应该说成有限整环,因为如  $\mathbb{Z}_6$  是有限环,但 2、3 均不为零因子,这与整环定义的非零因子相矛盾。——审校者注

## 第6章 数论

### 6.1 引言

像大整数分解、素性检测、开方求根、求解不同模数的联立同余方程组等,这类问题的运算在现代密码学中经常遇到。这些也是数论中令人感兴趣的题目。本章我们学习数论中的基本知识和算法,它们和现代密码学有着很重要的关系。

#### 6.1.1 本章概述

6.2节介绍同余和剩余类的基本概念和运算。6.3节介绍欧拉 $\phi$ 函数。6.4节给出费马定理、欧拉定理和拉格朗日定理的统一认识。6.5节引入二次剩余概念。6.6节介绍模一个整数的平方根算法。最后,6.7节将介绍布鲁姆整数。

### 6.2 同余和剩余类

在4.3.2.5节我们定义了模一个正整数 $n > 1$ 的同余系统,也研究了那样系统的一些性质。在这一节我们将研究同余系统的其他一些结果。

**定理 6.1** 对整数 $n > 1$ ,同余关系(模 $n$ )具有自反性、对称性和传递性。即对任意的 $a, b, c \in \mathbb{Z}$ ,有

- i)  $a \equiv a \pmod{n}$ ;
- ii) 如果  $a \equiv b \pmod{n}$ , 则  $b \equiv a \pmod{n}$ ;
- iii) 如果  $a \equiv b \pmod{n}$ ,  $b \equiv c \pmod{n}$ , 则  $a \equiv c \pmod{n}$ 。 □

满足定理 6.1 中三条性质的关系称为等价关系。大家都知道,一个集合上的等价关系把这个集合分成若干个等价类。令“ $\equiv_n$ ”表示模 $n$ 同余的等价关系。这个关系定义在集合 $\mathbb{Z}$ 上,因此它把 $\mathbb{Z}$ 恰好分成 $n$ 个等价类,每个类包含与某整数模 $n$ 同余的所有整数。把这 $n$ 个类表示成

$$\overline{0}, \overline{1}, \dots, \overline{n-1}$$

其中

$$\overline{a} = \{x \in \mathbb{Z} \mid x \pmod{n} \equiv a\} \quad (6.2.1)$$

我们将上面每一个集合均称为一个模 $n$ 剩余类。显然,我们可以将 $\mathbb{Z}_n$ 看做是

$$\mathbb{Z}_n = \{\overline{0}, \overline{1}, \dots, \overline{n-1}\} \quad (6.2.2)$$

另一方面,如果将 $\mathbb{Z}$ 看做是 $\mathbb{Z}$ 的一个(平凡)子集,陪集 $n\mathbb{Z}$ (5.2.1节中的定义 5.7)是所有 $n$ 的倍数的整数集合,即

$$n\mathbb{Z} = \{0, \pm n, \pm 2n, \dots\} \quad (6.2.3)$$

现在考虑下面的群运算为加法运算的商群(5.2.1节中的定义 5.8):

$$\mathbb{Z}/n\mathbb{Z} = \{x + n\mathbb{Z} \mid x \in \mathbb{Z}\} \quad (6.2.4)$$

如果我们用式(6.2.3)中的 $n\mathbb{Z}$ 将式(6.2.4)展开,得到

$$\begin{aligned} \mathbb{Z}/n\mathbb{Z} &= \{x + n\mathbb{Z} \mid x \in \mathbb{Z}\} \\ &= \{0 + \{0, \pm n, \pm 2n, \dots\}, \\ &\quad 1 + \{0, \pm n, \pm 2n, \dots\}, \\ &\quad 2 + \{0, \pm n, \pm 2n, \dots\}, \\ &\quad \dots, \\ &\quad (n-1) + \{0, \pm n, \pm 2n, \dots\}\} \\ &= \{\{0, \pm n, \pm 2n, \dots\}, \\ &\quad \{1, \pm n+1, \pm 2n+1, \dots\}, \\ &\quad \{2, \pm n+2, \pm 2n+2, \dots\}, \\ &\quad \dots, \\ &\quad \{(n-1), \pm n+(n-1), \pm 2n+(n-1), \dots\}\} \end{aligned} \quad (6.2.5)$$

在式(6.2.5)中仅有  $n$  个不同的元素,没有别的情况。例如

$$n + \{0, \pm n, \pm 2n, \dots\} = \{0, \pm n, \pm 2n, \dots\}$$

和

$$(n+1) + \{0, \pm n, \pm 2n, \dots\} = \{1, \pm n+1, \pm 2n+1, \dots\}$$

等。注意到式(6.2.1)关于  $\bar{a}$  的定义,比较式(6.2.2)和式(6.2.5),现在可以确切地知道对于  $n > 1$ ,有:

$$\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$$

$\mathbb{Z}/n\mathbb{Z}$ 是模  $n$  剩余类的标准表示法(事实上,就是定义),尽管习惯上都这样表示,但本书将用  $\mathbb{Z}_n$  代替  $\mathbb{Z}/n\mathbb{Z}$ 。

**定理 6.2** 对任意的  $a, b \in \mathbb{Z}$ , 定义剩余类  $\bar{a}$  和  $\bar{b}$  之间的加法和乘法运算为

$$\bar{a} + \bar{b} = \overline{a + b}, \quad \bar{a} \cdot \bar{b} = \overline{ab}$$

于是对任意的  $n > 1$ , 由“(mod  $n$ )”定义的映射  $f: \mathbb{Z} \rightarrow \mathbb{Z}_n$  是从  $\mathbb{Z}$  到  $\mathbb{Z}_n$  的一个同态映射。□

### 6.2.1 $\mathbb{Z}_n$ 中运算的同余性质

由下面的定理看到,从  $\mathbb{Z}$  到  $\mathbb{Z}_n$  的同态表明  $\mathbb{Z}_n$  中的运算(模  $n$  运算)具有  $\mathbb{Z}$  中的运算性质。

**定理 6.3** 对任意的整数  $n > 1$ , 如果  $a \equiv b \pmod{n}$  和  $c \equiv d \pmod{n}$ , 则  $a \pm c \equiv b \pm d \pmod{n}$ ,  $ac \equiv bd \pmod{n}$ 。

虽然由  $\mathbb{Z}$  到  $\mathbb{Z}_n$  的同态关系直接可以得到上述定理,但在这里我们仅运用  $\mathbb{Z}_n$  中的运算性质给出一个证明。

**证明** 如果  $n \mid a - b$  和  $n \mid c - d$ , 则  $n \mid (a \pm c) - (b \pm d)$ , 同样有

$$n \mid (a - b)(c - d) = (ac - bd) - b(c - d) - d(a - b), \text{ 故有 } n \mid (ac - bd). \quad \square$$

定理 6.3 给出的  $\mathbb{Z}_n$  中的运算性质称为同余性质, 这些性质表明在等式两边做相同的计算就导出一个新的等式。但是定理 6.3 没有给出除法运算。在  $\mathbb{Z}$  中的除法运算有下面的同余性质:

$$\text{任意 } d \neq 0, \text{ 由 } a = b \text{ 可得 } ad = bd \quad (6.2.6)$$

在  $\mathbb{Z}_n$  中, 对于除法, 相应的同余性质公式与公式 (6.2.6) 不完全相同。在给出这个公式之前, 首先在整数集  $\mathbb{Z}$  中对公式 (6.2.6) 做一下解释。我们可以设想把  $\mathbb{Z}$  看成  $\mathbb{Z}_n$ , 这里的  $n = \infty$ ,  $\infty$  可以被任何整数整除, 并且得到的结果仍然是  $\infty$ 。于是, 我们可以进一步想像, 式 (6.2.6) 的第一个等式在模  $\infty$  的情况下成立, 而公式 (6.2.6) 的第二个等式在模  $\infty/d$  的情况下成立。由于  $\infty/d = \infty$ , 式 (6.2.6) 中的两个等式相同。下面的引理表明,  $\mathbb{Z}$  中的除法同余性质能够嵌入到  $\mathbb{Z}_n$  中。

**定理 6.4** 假设整数  $n > 1, d \neq 0$ , 如果  $ad \equiv bd \pmod{n}$ , 则  $a \equiv b \pmod{\frac{n}{\gcd(d, n)}}$ 。

**证明** 令  $k = \gcd(d, n)$ , 则由  $n \mid (ad - bd)$  得到  $(n/k) \mid (d/k)(a - b)$ 。由于  $\gcd(d/k, n/k) = 1$ , 可以由  $(n/k) \mid (k/k)(a - b)$  推出  $(n/k) \mid (a - b)$ 。□

最后我们得出在  $\mathbb{Z}_n$  中的运算完全具有在整数集  $\mathbb{Z}$  中的运算性质。因此, 我们有

**推论 6.1** 如果  $f(x)$  是整数集  $\mathbb{Z}$  上的一个多项式, 并且  $a \equiv b \pmod{n}$ , 其中整数  $n > 1$ , 则  $f(a) \equiv f(b) \pmod{n}$ 。□

### 6.2.2 求解 $\mathbb{Z}_n$ 中的线性同余式

在定理 4.2 (见 4.3.2.5 节) 中, 我们定义了模  $n$  的乘法逆, 要使整数  $a$  存在乘法逆元, 即唯一的整数  $x < n$  满足  $ax \equiv 1 \pmod{n}$ , 当且仅当  $a$  满足  $\gcd(a, n) = 1$ 。下面的定理给出一般情况下线性同余方程可解的条件。

**定理 6.5** 假设整数  $n > 1$ , 同余式

$$ax \equiv b \pmod{n} \quad (6.2.7)$$

可解当且仅当  $\gcd(a, n) \mid b$ 。

**证明** 由定义 4.4 (见 4.3.2.5 节), 同余式 (6.2.7) 就是下面的线性方程

$$ax + kn = b \quad (6.2.8)$$

其中  $k$  是  $\mathbb{Z}$  中的某个整数。

( $\Rightarrow$ ) 假设式 (6.2.8) 成立。由于  $\gcd(a, n)$  整除等式的左边, 也一定整除等式的右边。

( $\Leftarrow$ ) 对于  $a$  和  $n$ , 运用扩展的欧几里得算法 (见算法 4.2), 我们可以计算

$$a\lambda + \mu n = \gcd(a, n)$$

由于  $b/\gcd(a, n)$  是整数, 用这个整数乘等式的两边, 我们得到式 (6.2.8) 和式 (6.2.7), 其中  $x = \frac{\lambda b}{\gcd(a, n)} \pmod{n}$  是一个解。□

容易验证, 给定式 (6.2.7) 的解  $x$ ,

$$x + \frac{ni}{\gcd(a, n)} \pmod{n}, i = 0, 1, 2, \dots, \gcd(a, n) - 1$$

是  $\gcd(a, n)$  个小于  $n$  的不同解。显然,  $\gcd(a, n) = 1$  是同余式 (6.2.8) 存在小于  $n$  的惟一解的充分条件。

**例 6.1** 由于  $\gcd(2, 10) = 2 \nmid 5$ , 同余式  $2x \equiv 5 \pmod{10}$  不可解。事实上, 等式的左边,  $2x$  一定是偶数, 而等式的右边,  $10k + 5$  只能是奇数, 于是尝试求解这样的同余式, 就是试图让奇数和偶数相等, 这显然是不可能的。

另一方面, 由于  $\gcd(6, 36) \mid 18$ , 同余式

$$6x \equiv 18 \pmod{36}$$

是可解的。这 6 个解分别是 3, 9, 15, 21, 27 和 33。  $\square$

**定理 6.6** 假设整数  $n > 1$ , 如果  $\gcd(a, n) = 1$ , 则对所有的  $b, i, j$  满足  $0 \leq i < j < n$ , 都有  $ai + b \not\equiv aj + b \pmod{n}$ 。

**证明** 假设相反,  $ai + b \equiv aj + b \pmod{n}$ 。则由定理 6.4, 我们有  $i \equiv j \pmod{n}$ , 这与  $0 \leq i < j < n$  矛盾。  $\square$

这条性质表明, 对于  $a, n$  满足  $\gcd(a, n) = 1$ ,  $ai + b \pmod{n} (i = 0, 1, \dots, n-1)$  是模  $n$  的完全剩余系, 即当  $i$  历遍  $\mathbb{Z}_n$  时,  $ai + b \pmod{n}$  历遍  $\mathbb{Z}_n$ 。

### 6.2.3 中国剩余定理

我们已经讨论了形如式(6.2.7)的单个线性同余式可解的条件。以后我们会经常遇到求解不同模的线性同余方程组的问题:

$$\begin{aligned} a_1 x &\equiv b_1 \pmod{n_1} \\ a_2 x &\equiv b_2 \pmod{n_2} \\ &\vdots \\ a_r x &\equiv b_r \pmod{n_r} \end{aligned} \tag{6.2.9}$$

其中  $a_i, b_i \in \mathbb{Z}$  且  $a_i \neq 0, i = 1, 2, \dots, r$ 。

要使上面的同余方程组可解, 显然首先需要每一个同余式都可解。于是, 对  $i = 1, 2, \dots, r$ , 令

$$d_i = \gcd(a_i, n_i)$$

由定理 6.5 可得  $d_i \mid b_i$ 。在这个条件下, 运用乘法运算(定理 6.3)和除法运算(定理 6.4)的同余性质, 把方程组(6.2.9)简化为下面等价的线性同余方程组:

$$\begin{aligned} x &\equiv c_1 \pmod{m_1} \\ x &\equiv c_2 \pmod{m_2} \\ &\vdots \\ x &\equiv c_r \pmod{m_r} \end{aligned} \tag{6.2.10}$$

其中对于  $i = 1, 2, \dots, r$ :

$$m_i = n_i / d_i$$

并且



$$c_i = (b_i/d_i)(a_i/d_i)^{-1} \pmod{m_i}$$

注意, 由于  $\gcd(a_i/d_i, m_i) = 1$ ,  $(a_i/d_i)^{-1} \pmod{m_i}$  是存在的(回顾 4.3.2.5 节的定理 4.2)。

由线性代数的知识, 方程组(6.2.10)可以用下面的向量空间形式表示:

$$A\vec{X} = \vec{C} \quad (6.2.11)$$

其中

$$A = \begin{pmatrix} \bar{1}_{m_1} & & & \\ & \bar{1}_{m_2} & & \\ & & \ddots & \\ & & & \ddots \\ & & & & \bar{1}_{m_r} \end{pmatrix} \quad (6.2.12)$$

$$\vec{X} = \begin{pmatrix} x \\ x \\ \vdots \\ \vdots \\ \vdots \\ x \end{pmatrix} \quad (6.2.13)$$

$$\vec{C} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ \vdots \\ \vdots \\ c_r \end{pmatrix} \quad (6.2.14)$$

注意到由于同余方程组(6.2.10)的第  $i$  ( $i = 1, 2, \dots, r$ ) 个模  $m_i$  的等式成立, 在矩阵  $A$  的对角线部分, 表示 1 模  $m_i$  的剩余类, 即对某个整数  $k_i$  ( $i = 1, 2, \dots, r$ ),

$$\bar{1}_{m_i} = k_i m_i + 1 \quad (6.2.15)$$

矩阵  $A$  的其余部分表示 0 模相应的模数[即  $k_i$  ( $i = 1, 2, \dots, r$ ), 在第  $i$  行的零元表示 0 模  $m_i$ ].

于是, 对任意给定的  $r$  维向量空间  $\vec{C}$ , 求解方程组(6.2.10)或它的线性空间形式(6.2.11), 归结为确定对角矩阵  $A$ , 或对于  $i = 1, 2, \dots, r$ , 找满足式(6.2.15)的 1 模  $m_i$  的剩余类。由线性代数的基本知识我们知道, 如果矩阵  $A$  存在, 由于  $A$  的对角线元素均不为零, 这是个满秩矩阵, 阶为  $r$ , 于是存在惟一的解。

当式(6.2.10)的模数两两互素时, 不难找到一个剩余类为 1 的方程组。这是由于以后将经常用到的中国剩余定理(CRT)。

**定理 6.7 中国剩余定理** 对于线性同余方程组(6.2.10), 如果  $\gcd(m_i, m_j) = 1$  ( $1 \leq i < j \leq r$ ), 则存在  $\bar{1}_{m_i}$ , 满足

$$\bar{1}_{m_i} \equiv 0 \pmod{m_j} \quad (6.2.16)$$

因此,存在  $x \in \mathbb{Z}_M$  是方程组(6.2.10)的一个根,其中  $M = m_1, m_2, \dots, m_r$ 。

**证明** 首先证明解的存在性,然后证明解的惟一性。

**存在性** 对每一个  $i = 1, 2, \dots, r$ ,  $\gcd(m_i, M/m_i) = 1$ 。由定理 4.2 (见 4.3.2.5 节),存在  $y_i \in \mathbb{Z}_{m_i}$ , 满足

$$(M/m_i) y_i \equiv 1 \pmod{m_i} \quad (6.2.17)$$

进一步,对于  $j \neq i$ , 由于  $m_j \mid (M/m_i)$ , 我们有

$$(M/m_i) y_i \equiv 0 \pmod{m_j} \quad (6.2.18)$$

于是,  $(M/m_i) y_i$  恰好是代表元  $\bar{1}_{m_i}$  数。令

$$x \leftarrow \sum_{i=1}^r \bar{1}_{m_i} c_i \pmod{M} \quad (6.2.19)$$

则  $x$  是方程组(6.2.10)的一个解,也是模  $M$  的一个剩余类。

**惟一性** 考虑由式(6.2.11)、式(6.2.12)、式(6.2.13)和式(6.2.14)定义的线性方程组,满足矩阵  $A$  的元素和向量  $\vec{C}$  都在  $\mathbb{Z}$  中(即都是整数)。注意到在  $\mathbb{Z}$  中,

$$\det(A) = \bar{1}_{m_1} \bar{1}_{m_2} \cdots \bar{1}_{m_r} \neq 0 \quad (6.2.20)$$

于是矩阵  $A$  的  $r$  列(向量)组成  $r$  维向量空间  $\underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_r$  的一组基(这个基与线性代数中所谓的“自然基”相似,在线性代数中的基向量中,仅有非零元素是 1)。所以,对任意向量  $\vec{C} \in \underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_r$ , 方程组(6.2.11)有惟一的解  $\vec{X} \in \underbrace{\mathbb{Z} \times \mathbb{Z} \times \cdots \times \mathbb{Z}}_r$ 。在存在性证明中,我们已经看到式(6.2.19)给出了  $\vec{X}$  的惟一元素。□

定理 6.7 的证明是构造性的,也就是说,我们已经构造了求解方程组(6.2.10)的一个算法。这个算法在算法 6.1 中具体描述。

在算法 6.1 中,仅有的耗时部分是第 2 步的(a),在这里计算大整数的乘法逆,可以应用扩展的欧几里得算法(算法 4.2)。对于  $i = 1, 2, \dots, r$ , 考虑到  $m_i < M$ , 算法 6.1 的时间复杂度是  $O_B(r(\log M)^2)$ 。

### 算法 6.1 中国剩余算法

输入 整数多元组  $(m_1, m_2, \dots, m_r)$ , 两两互素;

整数多元组  $(c_1 \pmod{m_1}, c_2 \pmod{m_2}, \dots, c_r \pmod{m_r})$ 。

输出 整数  $x < M = m_1, m_2, \dots, m_r$  满足方程组(6.2.10)。

1.  $M \leftarrow m_1 m_2 \cdots m_r$ ;

2. for ( $i$  from 1 to  $r$ ) do

(a)  $y_i \leftarrow (M/m_i)^{-1} \pmod{m_i}$  (\* 扩展欧几里得算法 \*);

(b)  $\bar{1}_{m_i} \leftarrow y_i M/m_i$ ;

3. return  $(\sum_{i=1}^r \bar{1}_{m_i} c_i \pmod{M})$ 。

由定理 6.7 很容易得到以下结果:

- i) 每一个  $x \in \mathbb{Z}_M$ , 可以对应一个向量  $\vec{C} \in \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$ 。由式(6.2.19)我们看到  $\vec{C}$  中的元素可以由  $c_i \leftarrow x \pmod{m_i}$  计算, 其中  $i = 1, 2, \cdots, r$ ;
- ii) 特别地,  $\mathbb{Z}_M$  中的 0 和 1 分别对应  $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$  中的  $\vec{0}$  和  $\vec{1}$ ;

$$\text{iii) } x, x' \text{ 分别对应 } \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_r \end{pmatrix}, \begin{pmatrix} c'_1 \\ c'_2 \\ \vdots \\ c'_r \end{pmatrix}, x \cdot x' \text{ 对应 } \begin{pmatrix} c_1 \cdot c'_1 \pmod{m_1} \\ c_2 \cdot c'_2 \pmod{m_2} \\ \vdots \\ c_r \cdot c'_r \pmod{m_r} \end{pmatrix}$$

于是, 我们已经证明了下面的定理(由定义 5.16):

**定理 6.8** 如果  $\gcd(m_i, m_j) = 1$ , 其中  $1 \leq i < j \leq r$ , 则对于  $M = m_1 m_2 \cdots m_r$ ,  $\mathbb{Z}_M$  同构于  $\mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$ , 并且这个同构

$$f: \mathbb{Z}_M \rightarrow \mathbb{Z}_{m_1} \times \mathbb{Z}_{m_2} \times \cdots \times \mathbb{Z}_{m_r}$$

是

$$f(x) = (x \pmod{m_1}, x \pmod{m_2}, \cdots, x \pmod{m_r}) \quad \square$$

在研究某个以合数为模的群上构造的密码体制或协议中, 定理 6.8 是很有用的。在本书的其余部分也将经常用到  $\mathbb{Z}_n^*$  与  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$  之间的同构, 其中  $n = pq$ ,  $p, q$  是素数。例如, 我们将用到下面一条性质: 非循环群  $\mathbb{Z}_n^*$  是由两个分别属于循环群  $\mathbb{Z}_p^*$  和  $\mathbb{Z}_q^*$  的生成元生成的。

现在看一下中国剩余定理的应用: 运用同构关系简化计算。

**例 6.2** 到目前为止, 我们还不知道如何求模一个整数的平方根(我们将在 6.6 节中研究具体的方法)。然而, 在某些情况下, 在某个集合中(如在  $\mathbb{Z}$  中), 平方数是显然的, 这样在这个集合中求平方根也是容易的, 不需要模运算。现在我们应用定理 6.8 计算 29 在  $\mathbb{Z}_{35}$  中的一个平方根。

仅由目前我们所学到的知识, 还不能容易地判断 29 是否为  $\mathbb{Z}_{35}$  里的一个平方数, 于是到目前为止我们还不知道如何直接求它的平方根。然而, 如果我们应用定理 6.8 把 29 映射到同构空间  $\mathbb{Z}_5 \times \mathbb{Z}_7$  中去, 我们有

$$29 \pmod{5} \mapsto 4, \quad 29 \pmod{7} \mapsto 1$$

即映像为 $(4, 1)$ 。显然 4 和 1 都是平方数, 2 是 4 的一个平方根, 1 是 1 的一个平方根。由同构性质,  $\mathbb{Z}_{35}$  中 29 的一个平方根对应于  $\mathbb{Z}_5 \times \mathbb{Z}_7$  中的  $(2, 1)$ 。应用中国剩余算法(算法 6.1), 我们得到

$$\bar{1}_5 = 21, \bar{1}_7 = 15$$

并且

$$\sqrt{29} \equiv 21 \cdot 2 + 15 \cdot 1 \equiv 22 \pmod{35}$$

的确,  $22^2 = 484 \equiv 29 \pmod{35}$ 。  $\square$

事实上, 29 在  $\mathbb{Z}_{35}^*$  中有 4 个不同的平方根。作为一个习题, 请读者求 29 的另外三个平方根(见习题 6.4)。

### 6.3 欧拉 $\phi$ 函数

5.2.3 节中的定义 5.11 定义了欧拉  $\phi$  函数。现在我们来研究它的一些有用的性质。

**引理 6.1** 设  $\phi(n)$  是定义 5.11 所定义的欧拉  $\phi$  函数。则

- i)  $\phi(1) = 1$ 。
- ii) 如果  $p$  是素数, 则  $\phi(p) = p - 1$ 。
- iii) 欧拉  $\phi$  函数是积性函数。即如果  $\gcd(m, n) = 1$ , 则  $\phi(mn) = \phi(m)\phi(n)$ 。
- iv) 如果  $n = p_1^{a_1} p_2^{a_2} \cdots p_k^{a_k}$  是  $n$  的素分解, 则

$$\phi(n) = n \left(1 - \frac{1}{p_1}\right) \left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$$

**证明** 由定义 5.11, (i) 和 (ii) 显然成立。

iii) 由于  $\phi(1) = 1$ , 当  $m = 1$  或  $n = 1$  时, 等式  $\phi(mn) = \phi(m)\phi(n)$  成立。于是假设  $m > 1$ ,  $n > 1$ 。对于  $\gcd(m, n) = 1$ , 考虑下面的排列

$$\begin{array}{cccccc} 0 & 1 & 2 & \cdots & m-1 & \\ m & m+1 & m+2 & \cdots & m+(m-1) & \\ \cdot & \cdot & \cdot & \cdots & \cdot & \\ \cdot & \cdot & \cdot & \cdots & \cdot & \end{array} \quad (6.3.1)$$

$$(n-1)m \quad (n-1)m+1 \quad (n-1)m+2 \quad \cdots \quad (n-1)m+(m-1)$$

一方面, 式(6.3.1)由  $mn$  个连续的整数组成, 于是它们就是模  $mn$  的所有整数, 这样就包含了  $\phi(mn)$  个与  $mn$  互素的元素。

另一方面, 对于式(6.3.1), 第一行是模  $m$  的所有元素, 并且任意一列的所有元素均模  $m$  同余。于是存在  $\phi(m)$  列包含了与  $m$  互素的所有整数。令

$$b, m+b, 2m+b, \cdots, (n-1)m+b$$

是其中任意一列中的  $n$  个元素。由于  $\gcd(m, n) = 1$ , 由定理 6.6, 这  $n$  个元素是模  $n$  的完全剩余系。于是在这样的一列元素中, 有  $\phi(n)$  个元素与  $n$  互素。于是我们得到, 在式(6.3.1)中存在  $\phi(m)\phi(n)$  个既与  $m$  互素又与  $n$  互素的元素。进而注意到任何元素既与  $m$  互素又与  $n$  互素当且仅当与  $mn$  互素。

总结以上两段的结果我们就得到  $\phi(mn) = \phi(m)\phi(n)$ 。

iv) 对任意素数  $p$ , 在  $1, 2, \dots, p^e$  中不与  $p^e$  互素的数是  $p$  的倍数, 即  $p, 2p, \dots, p^{e-1}p$ 。显然, 恰好存在  $p^{e-1}$  个这样的数。于是有

$$\phi(p^e) = p^e - p^{e-1} = p^e \left(1 - \frac{1}{p}\right)$$

这个结果对任意的素数幂  $p^e | n$  且  $p^{e+1} \nmid n$  都成立。注意到  $n$  的这些素数幂因子是两两互素的, 由 iii) 的结果可以证明。□

在 4.5 节我们考虑了所谓的无平方因子问题: 判断一个奇合数  $n$  是否为平方因子。我们用  $\phi(n)$  作为一个辅助输入来证明无平方因子问题属于  $\mathcal{NP}$ 。由引理 6.1 的性质 (iv), 我们得到对任意的  $p > 1$ , 如果  $p^2 | n$ , 则  $p | \phi(n)$ 。这就是为什么我们用  $\gcd(n, \phi(n)) = 1$  来作为  $n$  无平方因子的一个证明。读者可以考虑  $\gcd(n, \phi(n)) > 1$  的情况 [注意这种情况:  $n = pq$  且  $p | \phi(q)$ ], 见习题 6.5]。

欧拉  $\phi$  函数有以下很好的性质。

**定理 6.9** 对于整数  $n > 0$ ,  $\sum_{d|n} \phi(d) = n$ 。

**证明** 假设  $S_d = \{x | 1 \leq x \leq n, \gcd(x, n) = d\}$ 。显然, 对每个  $d | n$ , 集合  $S = \{1, 2, \dots, n\}$  被分割成不相交的子集  $S_d$ 。所以

$$\bigcup_{d|n} S_d = S$$

注意到对任意的  $d | n$ , 有  $\#S_d = \phi(n/d)$ , 于是

$$\sum_{d|n} \phi(n/d) = n$$

然而, 对任意的  $d | n$ , 我们有  $(n/d) | n$ , 所以

$$\sum_{d|n} \phi(n/d) = \sum_{(n/d)|n} \phi(n/d) = \sum_{d|n} \phi(d) \quad \square$$

**例 6.3** 对于  $n = 12$ ,  $d | 12$  可能的值是 1, 2, 3, 4, 6 和 12。我们有  $\phi(1) + \phi(2) + \phi(3) + \phi(4) + \phi(6) + \phi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12$ 。□

## 6.4 费马定理、欧拉定理、拉格朗日定理

在第 4 章我们引入了费马小定理 [同余式 (4.4.8)], 并且在前面已经用过几次, 到目前为止还未证明。现在我们通过证明它是数论中另外一个著名定理——欧拉定理的一种特殊情形来证明费马小定理。

**定理 6.10 费马小定理** 如果  $p$  是素数, 并且  $p \nmid a$ , 则  $a^{p-1} \equiv 1 \pmod{p}$ 。

由于  $\phi(p) = p - 1$  ( $p$  是素数), 费马小定理是下列定理的一种特殊情形。

**定理 6.11 欧拉定理** 如果  $\gcd(a, n) = 1$ , 则  $a^{\phi(n)} \equiv 1 \pmod{n}$ 。

**证明** 对于  $\gcd(a, n) = 1$ , 我们知道  $a \pmod{n} \in \mathbb{Z}_n^*$ , 并且  $\#\mathbb{Z}_n^* = \phi(n)$ 。由推论 5.2, 我们有  $\text{ord}_n(a) | \#\mathbb{Z}_n^*$ , 于是  $a^{\phi(n)} \equiv 1 \pmod{n}$ 。□

由于在定理 6.11 的证明中使用的推论 5.2 是拉格朗日定理(定理 5.1)的一个直接应用,于是我们可以认为,费马小定理和欧拉定理是拉格朗日定理的特殊情形。

在第 4 章我们看到了费马小定理在素性检测方面的重要应用,素性检测对于很多公钥密码体制和协议中密钥的生成是很有用的。欧拉定理在 8.5 节介绍的 RSA 密码体制中有重要的应用。

## 6.5 二次剩余

二次剩余在数论中扮演着很重要的角色。例如,整数分解算法都用到了二次剩余。二次剩余也经常用在加密和一些有趣的密码协议中。

**定义 6.1 二次剩余** 设整数  $n > 1$ 。对于  $a \in \mathbb{Z}_n^*$ ,  $a$  叫做模  $n$  的二次剩余,如果存在  $x \in \mathbb{Z}_n$ , 满足  $x^2 \equiv a \pmod{n}$ ; 否则,  $a$  就叫做模  $n$  的二次非剩余。用  $QR_n$  表示模  $n$  的二次剩余集合,用  $QNR_n$  表示模  $n$  的二次非剩余集合。

**例 6.4** 计算  $QR_{11}$ , 所有模 11 的二次剩余组成的集合。  $QR_{11} = \{1^2, 2^2, 3^2, 4^2, 5^2, 6^2, 7^2, 8^2, 9^2, 10^2\} \pmod{11} = \{1, 3, 4, 5, 9\}$ 。  $\square$

在上面例子中,我们通过对  $\mathbb{Z}_{11}^*$  中所有的数求平方得到了  $QR_{11}$ 。我们也可以不这样做。事实上,我们可以验证

$$QR_{11} = \{1^2, 2^2, 3^2, 4^2, 5^2\} \pmod{11}$$

即只要对不超过模数 1/2 的所有数求平方就足够了。对于模为任意素数的情形,下面的定理得到了这个结果。

**定理 6.12** 假设  $p$  是素数,则

- i)  $QR_p = \{x^2 \pmod{p} \mid 0 < x \leq (p-1)/2\}$ ;
- ii) 恰好存在  $(p-1)/2$  个模  $p$  二次剩余和  $(p-1)/2$  个模  $p$  二次非剩余,即  $\mathbb{Z}_p^*$  被分成大小相等的两个子集  $QR_p$  和  $QNR_p$ 。

**证明** i) 显然,集合  $S = \{x^2 \pmod{p} \mid 0 < x \leq (p-1)/2\} \subseteq QR_p$ 。要证明  $QR_p = S$ , 只需证明  $QR_p \subseteq S$ 。

假设任意的  $a \in QR_p$ , 则存在  $x < p$ , 满足  $x^2 \equiv a \pmod{p}$ 。如果  $x \leq (p-1)/2$ , 则  $a \in S$ 。假设  $x > (p-1)/2$ , 则  $y = p - x \leq (p-1)/2$  和  $y^2 \equiv (p-x)^2 \equiv p^2 - 2px + x^2 \equiv x^2 \equiv a \pmod{p}$ 。于是  $QR_p \subseteq S$ 。

ii) 要证明  $\#QR_p = (p-1)/2$ , 只需证明对于  $0 < x < y \leq (p-1)/2$ , 有  $x^2 \not\equiv y^2 \pmod{p}$ 。假设相反地,  $x^2 - y^2 \equiv (x+y)(x-y) \equiv 0 \pmod{p}$ , 则  $p \mid x+y$  或  $p \mid x-y$ 。由于  $x+y < p$ , 只有后者是可能的, 这样得到  $x = y$ , 矛盾。

由于  $\#QNR_p = \mathbb{Z}_p^* \setminus QR_p$  和  $\#\mathbb{Z}_p^* = p-1$ , 于是  $\#QNR_p = (p-1)/2$ 。  $\square$

在定理 6.12(i) 的证明中, 我们也已经证明了下面的结论:

**推论 6.2** 假设  $p$  为素数, 则对任意的  $a \in QR_p$ , 恰好存在  $a$  模  $p$  的两个平方根。用  $x$  表示其中的一个, 则另一个是  $-x (= p-x)$ 。  $\square$



### 6.5.1 二次剩余的判定

给定一个模数,我们经常需要判断一个数是否为二次剩余。这就是所谓的二次剩余判定问题。

**定理 6.13 欧拉准则** 设  $p$  为素数,则对任意的  $x \in \mathbb{Z}_p^*$ ,  $x \in \text{QR}_p$  当且仅当

$$x^{(p-1)/2} \equiv 1 \pmod{p} \quad (6.5.1)$$

**证明**  $(\Rightarrow)$  对于  $x \in \text{QR}_p$ , 存在  $y \in \mathbb{Z}_p^*$ , 满足  $y^2 \equiv x \pmod{p}$ 。于是由费马定理(定理 6.10),  $x^{(p-1)/2} \equiv y^{p-1} \equiv 1 \pmod{p}$ 。

$(\Leftarrow)$  假设  $x^{(p-1)/2} \equiv 1 \pmod{p}$ , 则  $x$  是多项式  $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$  的一个根。注意到  $\mathbb{Z}_p$  是一个域, 由定理 5.9(iii)(见 5.4.3 节), 域中的每个元素均为多项式  $y^p - y \equiv 0 \pmod{p}$  的根。换句话说, 域中的每个非零元素, 即群  $\mathbb{Z}_p^*$  中的每个元素, 均是

$$y^{p-1} - 1 \equiv (y^{(p-1)/2} - 1)(y^{(p-1)/2} + 1) \equiv 0 \pmod{p}$$

的根。

由于这个  $(p-1)$  次多项式至多有  $(p-1)$  个根, 所以所有的根是各不相同的。因此, 多项式  $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$  的  $(p-1)/2$  个根必定也各不相同。在定理 6.12 中我们已经证明  $\text{QR}_p$  恰好包含  $(p-1)/2$  个元素, 并且都满足  $y^{(p-1)/2} - 1 \equiv 0 \pmod{p}$ 。  $\mathbb{Z}_p^*$  中其余的任意元素必定满足  $y^{(p-1)/2} + 1 \equiv 0 \pmod{p}$ 。所以  $x \in \text{QR}_p$ 。  $\square$

在定理 6.13 的证明过程中, 我们已经证明如果  $x \in \mathbb{Z}_p$  不满足上面的准则, 则

$$x^{(p-1)/2} \equiv -1 \pmod{p} \quad (6.5.2)$$

欧拉准则提供了一个判断  $\mathbb{Z}_p^*$  中的一个元素是否为二次剩余的标准: 如果满足同余式(6.5.1), 则  $x \in \text{QR}_p$ ; 否则满足式(6.5.2)且  $x \in \text{QNR}_p$ 。

假设  $n$  是合数, 素分解为

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k} \quad (6.5.3)$$

于是, 由定理 6.8,  $\mathbb{Z}_n$  同构于  $\mathbb{Z}_{p_1^{e_1}} \times \mathbb{Z}_{p_2^{e_2}} \times \cdots \times \mathbb{Z}_{p_k^{e_k}}$ 。由于同构保持运算, 我们有:

**定理 6.14** 设  $n$  为合数, 有式(6.5.3)形式的完全分解。则  $x \in \text{QR}_n$  当且仅当  $x \pmod{p_i^{e_i}} \in \text{QR}_{p_i^{e_i}}$ , 于是当且仅当对任意的素数  $p_i, i=1, 2, \dots, k$ , 有  $x \pmod{p_i} \in \text{QR}_{p_i}$  成立。  $\square$

所以, 如果知道  $n$  的分解, 给定  $x \in \mathbb{Z}_n^*$ ,  $x$  模  $n$  的二次剩余就可以通过确定每个素数  $p \mid n$  的二次剩余来确定。后者可以通过验证欧拉准则来完成。

然而, 如果不知道  $n$  的分解, 确定模  $n$  的二次剩余是很困难的。

**定义 6.2 二次剩余判定(QR)问题**

输入  $n$ : 一个合数;

$x \in \mathbb{Z}_n^*$ 。

输出 是(如果  $x \in \text{QR}_n$ )。

QRP 是数论中一个公认的难题,也是在高斯的“算法专题”中讨论的四个主要算术问题之一[121]。对二次剩余问题的一个有效的解决方法将意味着解决数论中其他几个公开问题的有效方法。在第 14 章,我们将研究一种熟悉的公钥密码体制——Goldwasser-Micali 密码体制,这个密码体制的安全性基于确定 QRP 的困难问题。

结合定理 6.12 和定理 6.14 我们得到:

**定理 6.15** 设  $n$  为含有  $k$  个不同的素因子的合数,则恰好  $\mathbb{Z}_n^*$  中元素的  $\frac{1}{2^k}$  是模  $n$  的二次剩余。

于是,对于合数  $n$ ,一个判定模  $n$  二次剩余的有效算法将提供二次剩余在  $\mathbb{Z}_n^*$  中所占比例的一个有效统计检验,于是由定理 6.15,也提供了回答  $n$  是否有两个或三个不同素因子问题的有效算法。由定理 6.15,这是因为对于前一种情况( $n$  有两个不同的素因子),在  $\mathbb{Z}_n^*$  中恰有  $1/4$  的元素是二次剩余,而对于后一种情况,恰有  $1/8$  的元素是二次剩余。因此,能够区分集合  $E_{2\text{-Prime}}$  和  $E_{3\text{-Prime}}$  (见 4.7 节)。

目前,对于一个不知道分解的合数  $n$  来说,不存在能够确定模  $n$  二次剩余的  $n$  的多项式时间算法。

### 6.5.2 勒让德-雅可比符号

运用欧拉准则(6.5.1)检验以素数为模的二次剩余需要复杂的模指数运算。然而,二次剩余可以用更快的算法来检验。这样的算法基于勒让德-雅可比符号的定义。

**定义 6.3 勒让德-雅可比符号** 对于任意的素数  $p$  和任意的  $x \in \mathbb{Z}_p^*$ , 令

$$\left(\frac{x}{p}\right) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{若 } x \in \text{QR}_p \\ -1 & \text{若 } x \in \text{QNR}_p \end{cases}$$

$\left(\frac{x}{p}\right)$  叫做  $x$  模  $p$  的勒让德符号。

假设  $n = p_1 p_2 \cdots p_k$  是  $n$  的素分解(因子可重复)。则

$$\left(\frac{x}{n}\right) \stackrel{\text{def}}{=} \left(\frac{x}{p_1}\right) \left(\frac{x}{p_2}\right) \cdots \left(\frac{x}{p_k}\right)$$

叫做  $x$  模  $n$  的雅可比符号。

无论  $b$  是否为素数,在本书的其余部分均用  $\left(\frac{a}{b}\right)$  表示雅可比符号。

对于  $p$  是素数,比较式(6.5.1)、式(6.5.2)和定义 6.3,我们有

$$\left(\frac{x}{p}\right) = x^{(p-1)/2} \pmod{p} \quad (6.5.4)$$

另外,雅可比符号还有下面的性质:

**定理 6.16** 雅可比符号有以下性质:

- i)  $\left(\frac{1}{n}\right) = 1$ ;
- ii)  $\left(\frac{xy}{n}\right) = \left(\frac{x}{n}\right) \left(\frac{y}{n}\right)$ ;

$$\text{iii)} \left(\frac{x}{mn}\right) = \left(\frac{x}{m}\right) \left(\frac{x}{n}\right);$$

$$\text{iv)} \text{ 如果 } x \equiv y \pmod{n}, \text{ 则 } \left(\frac{x}{n}\right) = \left(\frac{y}{n}\right);$$

(以下的  $m$  和  $n$  都是奇数)

$$\text{v)} \left(\frac{-1}{n}\right) = (-1)^{(n-1)/2};$$

$$\text{vi)} \left(\frac{2}{n}\right) = (-1)^{(n^2-1)/8};$$

$$\text{vii)} \text{ 如果 } \gcd(m, n) = 1 \text{ 且 } m, n > 2, \text{ 则 } \left(\frac{m}{n}\right) \left(\frac{n}{m}\right) = (-1)^{(m-1)(n-1)/4}.$$

定理 6.16 的(i ~ iv)可由雅可比符号直接得到,关于(v ~ vii)的证明也没有用到特殊的方法。但是,由于证明过程冗长并且与本书的主题没有太大的关系,我们在这里省略证明,推荐读者参阅关于数论的文献([172],[178])。

定理 6.16 的(vii)称为高斯二次互反律。由于有了这个互反律,不难发现对于  $\gcd(x, n) = 1$ , 有了一个计算  $\left(\frac{x}{n}\right)$  的方法,因此,与计算最大公因子的复杂度相同。

**注释 6.1** 当我们应用定理 6.16 计算雅可比符号时,计算(v-vii)右边的值一定不要用指数运算。由于  $\text{ord}(-1) = 2$  (乘法),我们所需要的只是这些指数的奇偶性。在算法 6.2 中,我们通过验证 2 是否整除指数来实现运算。□

算法 6.2 对定理 6.16 列出的雅可比符号的性质以递归的形式给予了详细说明。

#### 算法 6.2 勒让德/雅可比符号

输入 奇数  $n > 2$ , 整数  $x \in \mathbb{Z}_n^*$ 。

输出  $\left(\frac{x}{n}\right)$ 。

Jacobi( $x, n$ )

1. 如果( $x = 1$ )返回(1);

2. 如果( $2 \mid x$ )

(a) 如果( $2 \mid (n^2 - 1)/8$ )返回(Jacobi( $x/2, n$ ));

(b) 返回( $-\text{Jacobi}(x/2, n)$ );

(\* 现在  $x$  是奇数 \*)

3. 如果( $2 \mid (x-1)(n-1)/4$ )返回(Jacobi( $n \bmod x, x$ ));

4. 返回( $-\text{Jacobi}(n \bmod x, x)$ )。

在算法 6.2 中,函数 Jacobi(,)的每一次递归将会得到或者是第一次输入除以 2 后的值,或者是模上第一次输入的值。于是,至多  $\log_2 n$  次调用后,第一次输入的值将会约简为 1,达到了结束的条件。这样,由于每一步模运算需要  $O_B((\log n)^2)$  时间,严格地说,算法 6.2 可以在  $O_B((\log n)^3)$  时间内计算  $\left(\frac{x}{n}\right)$ 。

然而,应该注意到,为了使算法更容易理解,我们又一次选择了牺牲效率。

通过仔细地实现,步骤 3、4 的总共运算在  $O_B((\log n)^2)$  次以内,而不是每一步模运算在  $O_B((\log n)^2)$  次以内。这种情形与使用仔细设计的算法来计算最大公因子时的情形恰好相同:运用式(4.3.12)表述的事实。因此,对于  $x \in \mathbb{Z}_n^*$ ,可以在  $O_B((\log n)^2)$  时间内计算  $(\frac{x}{n})$ 。在文献[80]的第 1 章里介绍了一个类似于算法 6.2 的详细算法。

5.4.5 节的欧拉准则需要  $O_B((\log p)^3)$  次模指数运算,这样与欧拉准则相比,用算法 6.2 验证模数为素数  $p$  时的二次剩余问题要快  $\log p$  倍。

**例 6.5** 证明  $384 \in \text{QNR}_{443}$ 。

运用算法 6.2,我们有

$$\begin{aligned} \text{Jacobi}(384, 443) &= -\text{Jacobi}(192, 443) \\ &= \text{Jacobi}(96, 443) \\ &= -\text{Jacobi}(48, 443) \\ &= \text{Jacobi}(24, 443) \\ &= -\text{Jacobi}(12, 443) \\ &= \text{Jacobi}(6, 443) \\ &= -\text{Jacobi}(3, 443) \\ &= \text{Jacobi}(2, 3) \\ &= -\text{Jacobi}(1, 3) \\ &= -1 \end{aligned}$$

所以  $384 \in \text{QNR}_{443}$ 。 □

最后,我们应该注意到,用算法 6.2 计算雅可比符号不需要分解  $n$ 。这是一个很重要的性质,它广泛应用于公钥密码学当中,如 Goldwasser-Micali 密码体制(见 14.3.3 节)、Blum 掷币协议(见第 19 章)。

## 6.6 模一个整数的平方根

在例 6.2 中,我们计算出了模一个整数的平方根。然而所用到的“算法”不能称为一个真正的算法,这是由于我们碰巧能够运用定理 6.8 中的同构,成功地把一个看起来困难的问题映射成两个容易的问题:计算 1 和 4 的平方根,它们恰好是整数  $\mathbb{Z}$  中的平方数,并且即使小学生也知道“求根算法”。通常情况下,对我们来说定理 6.8 的同构并不太好:在绝大多数情况下,同构映射的像并不是  $\mathbb{Z}$  中的平方数。

现在我们介绍模为一个正整数时,求一个二次剩余元素平方根的算法。首先,我们考虑模为一个素数时的情况。由推论 6.2,二次剩余元素的两个根在模这个素数的情况下互为补元;于是我们只需要计算二次剩余元素的其中一个平方根即可。

这个计算对于大多数的奇素数来说是比较容易的。这样的素数  $p$  包括  $p \equiv 3, 5, 7 \pmod{8}$ 。

### 6.6.1 求模为素数时的平方根

情况  $p \equiv 3, 7 \pmod{8}$

在这种情况下,  $p+1$  被 4 整除。对于  $a \in \text{QR}_p$ , 令

$$x \stackrel{\text{def}}{=} a^{(p+1)/4} \pmod{p}$$

于是, 由于  $a^{(p-1)/2} \equiv 1 \pmod{p}$ , 我们有

$$x^2 \equiv a^{(p+1)/2} \equiv a^{(p-1)/2} a \equiv a \pmod{p}$$

这样,  $x$  的确是  $a$  模  $p$  的一个平方根。

**情况  $p \equiv 5 \pmod{8}$**

在这种情况下,  $p+3$  被 8 整除;  $(p-1)/2$  是偶数,  $-1$  是一个二次剩余并且满足欧拉准则。对于  $a \in \text{QR}_p$ , 令

$$x \stackrel{\text{def}}{=} a^{(p+3)/8} \pmod{p} \quad (6.6.1)$$

由  $a^{(p-1)/2} \equiv 1 \pmod{p}$ , 得到  $a^{(p-1)/4} \equiv \pm 1 \pmod{p}$ 。事实上, 在域  $\mathbb{Z}_p^*$  中, 1 仅有两个平方根: 1 和  $-1$ 。因此

$$x^2 \equiv a^{(p+3)/4} \equiv a^{(p-1)/4} a \equiv \pm a \pmod{p}$$

这样, 式(6.6.1)计算的  $x$  就是  $a$  或  $-a$  的平方根。如果符号为“+”, 则我们就求得了一个平方根。如果符号为“-”, 我们有

$$-x^2 \equiv (\sqrt{-1}x)^2 \equiv a \pmod{p}$$

因此,

$$x \stackrel{\text{def}}{=} \sqrt{-1} a^{(p+3)/8} \pmod{p} \quad (6.6.2)$$

就是要求的根。于是求根归结为计算  $\sqrt{-1} \pmod{p}$ 。令  $b$  为  $\text{mod } p$  的任意一个二次非剩余。则由欧拉准则有

$$(b^{(p-1)/4})^2 \equiv b^{(p-1)/2} \equiv -1 \pmod{p}$$

于是  $b^{(p-1)/4} \pmod{p}$  可用来代替  $\sqrt{-1}$ 。由于

$$p^2 - 1 = (p+1)(p-1) = (8k+6)(8k+4) = 8(4k'+3)(2k''+1)$$

并且等式的右边为 8 与某个奇数的乘积。于是, 由定理 6.16(vi),  $2 \in \text{QNR}_p$ 。也就是说, 在这种情况下, 我们可以用  $2^{(p-1)/4}$  代替  $\sqrt{-1}$ 。于是, 不难验证式(6.6.2)就是

$$2^{(p-1)/4} a^{(p+3)/8} \equiv (4a)^{(p+3)/8} / 2 \pmod{p} \quad (6.6.3)$$

我们运用等式(6.6.3)的右边可以省去一项模指数运算。

算法 6.3 的时间复杂度为  $O_B((\log p)^3)$ 。

### 算法 6.3 模素数的平方根 [ $p \equiv 3, 5, 7 \pmod{8}$ ] 的特殊情况]

输入 素数  $p$  满足  $p \equiv 3, 5, 7 \pmod{8}$ ; 整数  $a \in \text{QR}_p$ 。

输出  $a$  模  $p$  的一个平方根。

1. 如果  $(p \equiv 3, 7 \pmod{8})$  返回  $(a^{(p+1)/4} \pmod{p})$ ;  
(\* 以下  $p \equiv 5 \pmod{8}$  \*)
2. 如果  $(a^{(p+1)/4} \equiv 1 \pmod{p})$  返回  $(a^{(p+3)/8} \pmod{p})$ ;
3. 返回  $((4a)^{(p+3)/8} / 2)$ 。

### 求模为一般素数时的平方根

这里描述的算法是 Shanks 给出的(参阅文[80]的 1.5.1 节)。

对于一般的素数  $p$ , 我们可以写成

$$p-1=2^e q$$

其中  $q$  为奇数,  $e \geq 1$ 。由定理 5.2(见 5.2.3 节), 循环群  $\mathbb{Z}_p^*$  有惟一一个阶为  $2^e$  的循环子群  $G$ 。显然, 由于  $G$  中的二次剩余的阶整除  $2^{e-1}$ , 则  $G$  中的二次剩余的阶为 2 的幂的形式。对于  $a \in \text{QR}_p$ , 由于

$$a^{(p-1)/2} \equiv (a^q)^{2^{e-1}} \equiv 1 \pmod{p}$$

则  $a^q \pmod{p} \in G$  也是一个二次剩余。于是存在一个偶数  $k (0 \leq k < 2^e)$ , 满足

$$a^q g^k \equiv 1 \pmod{p} \quad (6.6.4)$$

其中  $g$  是  $G$  的生成元。假设我们已经找到了这个生成元  $g$  和偶数  $k$ , 则令

$$x \stackrel{\text{def}}{=} a^{(q+1)/2} g^{k/2}$$

容易验证  $x^2 \equiv a \pmod{p}$ 。

于是, 求  $a$  模  $p$  的二次剩余只需要解决下面两个问题: (i) 找出群  $G$  的一个生成元  $g$ , (ii) 求满足式(6.6.4)的最小的非负偶数  $k$ 。

问题(i)是很容易的。对于  $f \in \text{QNR}_p$ , 由于  $q$  是奇数,  $f^q \in \text{QNR}_p$ , 并且  $\text{ord}_p(f^q) = 2^e$ , 所以  $f^q$  是  $G$  的一个生成元。很容易找到这样的  $f$ : 随机选择  $f \in \mathbb{Z}_p^*$ , 验证  $\left(\frac{f}{p}\right) = -1$  (运用算法 6.2)。由于  $\mathbb{Z}_p^*$  中的一半元素是二次非剩余, 每一次找到这样的  $f$  的概率是 1/2。

问题(ii)也比较容易。由于  $G$  中非单位的二次剩余的阶为 2 的幂, 由式(6.6.4)很快就能找到  $k$ 。这样, 令

$$b \stackrel{\text{def}}{=} a^q \equiv a^q g^{2^e} \pmod{p} \quad (6.6.5)$$

则  $b \in G$ 。我们可以找最小的整数  $m (0 \leq m < e)$ , 满足

$$b^{2^m} \equiv 1 \pmod{p} \quad (6.6.6)$$

接着更改  $b$  的值

$$b \leftarrow b g^{2^{e-m}} \equiv a^q g^{2^{e-m}} \pmod{p} \quad (6.6.7)$$

注意, 式(6.6.7)中更改  $b$  值后,  $b$  的阶比式(6.6.5)中  $b$  的阶减小了, 但仍然是  $G$  中的一个二次剩余, 阶仍然是 2 的幂。因此, 减小的量一定是 2 的幂的形式。于是, 重复式(6.6.6)和式(6.6.7), 式(6.6.6)中的  $m$  就会减小。如果  $m=0$ , 式(6.6.6)表明  $b=1$ , 并且式(6.6.7)就是式(6.6.4), 所以, 通过在每一个循环中累加  $2^m$  就可以找到  $k$ 。这个搜索至多需要  $e$  次循环。

现在可以直接把我们的描述写成算法 6.4。

由于  $e < \log_2 p$ , 算法 6.4 的时间复杂度是  $O_B((\log p)^4)$ 。

#### 算法 6.4 模素数的平方根(一般情况)

输入 素数  $p$ ; 整数  $a \in \text{QR}_p$ 。

输出  $a \pmod{p}$  的一个平方根。

1. (\* 初始化 \*)  
令  $p-1=2^e q$ , 其中  $q$  为奇数;  $b \leftarrow a^q \pmod{p}$ ;  $r \leftarrow e$ ;  $k \leftarrow 0$ ;
2. (\* 问题(i), 运用算法 6.2 \*)  
找  $f \in \text{QNR}_p$ ;  $g \leftarrow f^q \pmod{p}$ ;
3. (\* 问题(ii), 搜索偶指数  $k$  \*)  
while( $b \neq 1$ ) do  
3.1 找最小的非负整数  $m$ , 满足  $b^{2^m} \equiv 1 \pmod{p}$ ;  
3.2  $b \leftarrow b g^{2^{r-m}} \pmod{p}$ ;  $k \leftarrow k + 2^{r-m}$ ;  $r \leftarrow m$ ;
4. 返回( $a^{(q+1)/2} g^{k/2} \pmod{p}$ )。

**注释 6.2** 为了更清楚地阐明问题, 我们在解释 Shanks 算法原理之后, 给出了算法 6.4; 特别地, 我们严格按照对问题(ii)的解释给出找偶指数  $k$  的算法。在这样做时, 我们对 Shanks 算法的描述牺牲了一点效率: 准确地找到  $k$ , 但在第 4 步中额外增加了一个模指数运算; 而由于可以在第 3 步中作为一个附属结果得到, 所以是不必要的。关于优化的 Shanks 算法, 可参阅文[80]的算法 1.5.1。□

最后我们要指出的是, 算法 6.4 包含算法 6.3 作为三个特殊情形。

### 6.6.2 求模为合数时的平方根

由定理 6.8, 对于  $n = pq$ , 其中  $p, q$  为素数,  $\mathbb{Z}_n^*$  同构于  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$ 。由于同构关系保持运算, 所以

$$x^2 \equiv y \pmod{n}$$

成立当且仅当对模为素数  $p$  和  $q$  时成立。因此, 如果给出  $n$  的分解, 用算法 6.5 可以求得模  $n$  的平方根。

显然算法 6.5 的时间复杂度是  $O_B((\log n)^4)$ 。

由推论 6.2,  $y \pmod{p}$  有两个不同的平方根, 分别记为  $x_p$  和  $p - x_p$ ; 对于  $y \pmod{q}$  同样也有两个, 记为  $x_q$  和  $q - x_q$ 。由  $\mathbb{Z}_n^*$  和  $\mathbb{Z}_p^* \times \mathbb{Z}_q^*$  之间的同构关系(定理 6.8), 我们知道  $y \in \text{QR}_n$  在  $\mathbb{Z}_n^*$  中有 4 个平方根。由算法 6.5, 这 4 个根是

$$\left. \begin{aligned} x_1 &\equiv \bar{1}_p x_p & + \bar{1}_q x_q \\ x_2 &\equiv \bar{1}_p x_p & + \bar{1}_q (q - x_q) \\ x_3 &\equiv \bar{1}_p (p - x_p) & + \bar{1}_q x_q \\ x_4 &\equiv \bar{1}_p (p - x_p) & + \bar{1}_q (q - x_q) \end{aligned} \right\} \pmod{n} \quad (6.6.8)$$

因此, 如果算法 6.5 的第二步应用式(6.6.8), 我们可以计算输入元素所有的 4 个根。

#### 算法 6.5 模一个合数的平方根

输入 素数  $p$  和  $q$ , 满足  $n = pq$ , 整数  $y \in \text{QR}_n$ 。

输出  $y$  模  $n$  的一个平方根。



1.  $x_p \leftarrow \sqrt{y \pmod p}$ ;  
 $x_q \leftarrow \sqrt{y \pmod q}$ ; (\* 应用算法 6.3 或 6.4 \*)
2. 返回  $(\bar{1}_p x_p + \bar{1}_q x_q \pmod n)$ 。 (\* 应用算法 6.1 \*)

作为一个练习,问:如果  $n = pqr$ , 其中  $p, q, r$  是不同的素数, 对于任意的  $y \in \text{QR}_n$ , 有几个平方根?

现在我们知道, 如果已知  $n$  的分解, 则对任意给定的  $\text{QR}_n$  中的元素, 可以有效地计算它的平方根。但是, 如果不知道  $n$  的分解, 关于模  $n$  的平方根我们能得到些什么? 下面定理的第三部分回答了这个问题。

**定理 6.17** 设  $n = pq$ , 其中  $p, q$  为两个不同的素数, 且  $y \in \text{QR}_n$ 。则由式(6.6.8)求得  $y$  的 4 个平方根有以下性质:

- i) 这 4 个根各不相同;
- ii)  $x_1 + x_4 = x_2 + x_3 = n$ ;
- iii)  $\gcd(x_1 + x_2, n) = \gcd(x_3 + x_4, n) = q, \gcd(x_1 + x_3, n) = \gcd(x_2 + x_4, n) = p$ 。

**证明**

- i) 注意到式(6.2.15)和式(6.2.16)定义的  $\bar{1}_p$  和  $\bar{1}_q$  的含义, 我们有  $x_1 \pmod q = x_q$  和  $x_2 \pmod q = q - x_q$ 。由于  $x_q$  和  $q - x_q$  是  $y \pmod q$  的两个不同的平方根, 所以  $x_1 \not\equiv x_2 \pmod n$  意味着  $x_1 \not\equiv x_2 \pmod q$ , 也就是说,  $x_1$  和  $x_2$  是不同的。其他情况也可以类似证明。
- ii) 由式(6.6.8), 我们有

$$x_1 + x_4 = x_2 + x_3 = \bar{1}_p p + \bar{1}_q q$$

等式右边的值模  $p$  和模  $q$  同余于 0。由于这些根都属于  $\mathbb{Z}_n^*$ , 我们有  $0 < x_1 + x_4 = x_2 + x_3 < 2n$ 。显然,  $n$  是区间  $(0, 2n)$  中模  $p$  和模  $q$  同余于 0 的惟一值, 于是  $x_1 = n - x_4$  和  $x_2 = n - x_3$ 。

- iii) 我们仅研究了  $x_1 + x_2$  这种情形; 其余的情形类似。由式(6.6.8), 我们有

$$x_1 + x_2 = 2 \cdot \bar{1}_p x_p + \bar{1}_q q$$

所以,  $x_1 + x_2 \pmod p \equiv 2x_p \not\equiv 0$  和  $x_1 + x_2 \equiv 0 \pmod q$ 。即  $x_1 + x_2$  不为 0 且是  $q$  的倍数, 但不是  $p$  的倍数。这就意味着  $\gcd(x_1 + x_2, n) = q$ 。□

假设存在有效的算法  $A$ , 输入  $(y, n)$  ( $y \in \text{QR}_n$ ), 输出  $x$  满足  $x^2 \equiv y \pmod n$ 。则我们可以运行  $A(x^2, n)$  求得  $x^2$  的一个平方根, 记为  $x'$ 。由定理 6.17 (iii), 满足  $1 < \gcd(x + x', n) < n$  的概率恰好是  $1/2$  (概率空间为  $y$  的 4 个根)。即算法  $A$  是分解  $n$  的一个有效算法。

结合算法 6.5 和定理 6.17(iii), 我们有

**推论 6.3** 设  $n = pq$ , 其中  $p$  和  $q$  是不同的奇素数, 则分解  $n$  计算上等价于求模  $n$  的平方根。□

由于  $n$  为奇数, 由定理 6.17(ii), 我们有

**推论 6.4** 设  $n = pq$ , 其中  $p$  和  $q$  是不同的奇素数, 则对任意的  $y \in \text{QR}_n$ ,  $y$  的两个平方根小于  $n/2$ , 另外两个根大于  $n/2$ 。□

## 6.7 Blum 整数

Blum 整数广泛地用于公钥密码学中。

**定义 6.4 Blum 整数** 一个合数  $n$  称为 Blum 整数,条件是  $n = pq$ ,其中  $p$  和  $q$  是两个不同的素数并满足  $p \equiv q \equiv 3 \pmod{4}$ 。

Blum 整数有很多有趣的性质。下面就是一些在公钥密码学和密码协议中非常有用的有趣性质。

**定理 6.18** 设  $n$  为 Blum 整数,则下面的性质成立:

- i)  $\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$  (所以  $\left(\frac{-1}{n}\right) = 1$ );
- ii) 对于  $y \in \mathbb{Z}_n^*$ , 如果,  $\left(\frac{y}{n}\right) = 1$  则  $y \in QR_n$  或者  $y \in QR_n$ , 或者  $-y = n - y \in QR_n$ ;
- iii) 任意  $y \in QR_n$  有 4 个根  $u, -u, v, -v$ , 并且都满足
  - a)  $\left(\frac{u}{p}\right) = 1, \left(\frac{u}{q}\right) = 1$ , 即  $u \in QR_n$ ;
  - b)  $\left(\frac{-u}{p}\right) = 1, \left(\frac{-u}{q}\right) = -1$ ;
  - c)  $\left(\frac{v}{p}\right) = -1, \left(\frac{v}{q}\right) = 1$ ;
  - d)  $\left(\frac{-v}{p}\right) = 1, \left(\frac{-v}{q}\right) = -1$ ;
- iv) 函数  $f(x) = x^2 \pmod{n}$  是集合  $QR_n$  的一个置换;
- v) 对于任意的  $y \in QR_n$ ,  $y$  的平方根中恰好有一个小于  $n/2$  且雅可比符号为 1;
- vi)  $\mathbb{Z}_n^*$  被分为 4 个等价类: 一个为乘群  $QR_n$ , 另外 3 个是陪集  $(-1)QR_n, \xi QR_n, (-\xi)QR_n$ ; 其中,  $\xi$  是 1 的平方根且雅可比符号为  $-1$ 。

**证明**

i) 注意到  $p \equiv 3 \pmod{4}$ , 于是  $\frac{p-1}{2} = 2k+1$ 。则由欧拉准则(6.5.1)我们有

$$\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}} = (-1)^{2k+1} = -1$$

类似地,  $\left(\frac{-1}{q}\right) = -1$ 。

ii) 由  $\left(\frac{y}{n}\right) = 1$  得到  $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = 1$ , 或  $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$ 。对于第一种情况, 由勒让德符号(定义 6.3)和定理 6.14,  $y \in QR_n$ 。对于第二种情况, (i) 意味着  $\left(\frac{-y}{p}\right) = \left(\frac{-y}{q}\right) = 1$ , 所以  $-y \in QR_n$ 。

iii) 首先, 由定理 6.17(ii), 我们可以把  $x$  的 4 个不同的平方根表示为  $u, -u (= n - u), v$  和  $-v$ 。

其次, 由  $u^2 \equiv v^2 \pmod{n}$ , 我们有  $(u+v)(u-v) \equiv 0 \pmod{p}$ , 即  $u \equiv \pm v \pmod{p}$ 。同

理,  $u \equiv \pm v \pmod{q}$ 。然而, 由定理 6.17(i),  $u \not\equiv \pm v \pmod{n}$ , 于是仅有下面的两种情形是可能的:

$$u \equiv v \pmod{p} \text{ 和 } u \equiv -v \pmod{q}$$

或

$$u \equiv -v \pmod{p} \text{ 和 } u \equiv v \pmod{q}$$

这两种情形连同(i)得到  $\left(\frac{u}{n}\right) = -\left(\frac{v}{n}\right)$ 。

所以, 如果  $\left(\frac{u}{n}\right) = 1$ , 则  $\left(\frac{v}{n}\right) = -1$ ; 如果  $\left(\frac{u}{n}\right) = -1$ , 则  $\left(\frac{v}{n}\right) = 1$ 。不失一般性, 由勒让德-雅可比符号的乘法性质和(i)可得到(a)~(d)中四个不同的勒让德符号。

iv) 对于任意  $y \in \text{QR}_n$ , 由(iii), 存在惟一的  $x \in \text{QR}_n$  满足  $f(x) = y$ 。所以,  $f(x)$  是一一映射, 即是集合  $\text{QR}_n$  的一个置换。

v) 由(iii), 雅可比符号为 1 的平方根是  $u$  或  $n-u$ 。由于  $n$  是奇数, 仅有一个小于  $n/2$  (于是, 恰好有一个平方根小于  $n/2$  且雅可比符号为  $-1$ , 另外两个大于  $n/2$ , 并且符号互反)。

vi) 不难验证,  $\text{QR}_n$  在模  $n$  乘法运算下构成一个群, 1 是单位元。由(iii), 1 的 4 个不同平方根在(a)、(b)、(c)和(d)中分别有 4 个不同的勒让德符号描述。因此, 这 4 个集合  $\text{QR}_n, (-1)\text{QR}_n, \xi\text{QR}_n, (-\xi)\text{QR}_n$  两两不相交。由定理 6.15, 这 4 个集合组成  $\mathbb{Z}_n^*$ ,  $\#\text{QR}_n = \frac{\#\mathbb{Z}_n^*}{4}$ 。□

## 6.8 本章小结

在这一章, 我们研究了关于初等数论的以下一些论题:

- 线性同余
- 中国剩余定理(含算法)
- 拉格朗日定理、欧拉定理和费马定理
- 二次剩余和勒让德-雅可比符号(含算法)
- 模为整数时的平方根和与整数分解的关系(含求根算法)
- Blum 整数和性质

除了介绍基本的知识和事实以外, 我们也研究了几个重要的算法(中国剩余、雅可比符号、求平方根), 并阐明了算法原理、分析了时间复杂度。同时, 我们认为这些算法不仅有重要的理论意义, 而且也有重要的实用价值: 它们经常用于密码学和密码协议中。

在本书的其余部分, 我们将经常用到本章学到的知识、事实、技巧和算法。

## 习题

- 6.1 设  $m, n$  为正整数满足  $m \mid n$ , 证明“模  $m$ ”运算把  $\mathbb{Z}_n$  分成  $n/m$  个等价类, 每个等价类有  $m$  个元素。
- 6.2 与上一问题条件相同, 证明  $\mathbb{Z}_n / m\mathbb{Z}_n = \mathbb{Z}_m$ 。
- 6.3 运用中国剩余算法(算法 6.1)构造  $\mathbb{Z}_{35}$  中的一个元素, 这个元素在定理 6.8 的同构映射下映射到  $(2, 3) \in \mathbb{Z}_5 \times \mathbb{Z}_7$ 。证明这个元素取得最大阶。

6.4 运用例 6.2 的方法在  $\mathbb{Z}_{35}^*$  中求 29 的其余 3 个平方根。用相似的方法在  $\mathbb{Z}_{35}^*$  中求 1 的 4 个平方根。

提示:  $29 \pmod{5} = 4$  有根 2 和  $3 (= -2 \pmod{5})$ ,  $29 \pmod{7} = 1$  有根 1 和  $6 (= -1 \pmod{7})$ ; 29 模 35 的 4 个根同构于  $\mathbb{Z}_5 \times \mathbb{Z}_7$  中的  $(2, 1)$ 、 $(2, 6)$ 、 $(3, 1)$  和  $(3, 6)$ 。

6.5 构造一个奇合数  $n$ , 满足  $n$  无平方因子, 即不存在素数  $p$  满足  $p^2 \mid n$ , 但是却有  $\gcd(n, \phi(n)) > 1$ 。

6.6 设  $m \mid n$ , 证明对任意  $x \in \mathbb{Z}_n^*$ , 都有  $\text{ord}_m(x) \mid \text{ord}_n(x)$ 。

6.7 设  $n = pq$ , 其中  $p, q$  是不同的素数。由于  $p-1 \mid \phi(n)$ , 在  $\mathbb{Z}_n^*$  中存在阶整除  $p-1$  的元素(类似地, 存在阶整除  $q-1$  的元素)。证明, 对任意的  $g \in \mathbb{Z}_n^*$ , 如果  $\text{ord}_n(g) \mid p-1$ , 并且  $\text{ord}_n(g) \nmid q-1$ , 则  $\gcd(g-1, n) = q$  (类似地, 任意  $h \in \mathbb{Z}_n^*$ ,  $\text{ord}_n(h) \mid q-1$ , 并且  $\text{ord}_n(h) \nmid p-1$ , 则  $\gcd(h-1, n) = p$ )。

6.8 设  $n = pq$ , 其中  $p, q$  是不同的素数。对任意  $g \in \mathbb{Z}_p^*$ , 等式  $g^{p+q} \equiv g^{n+1} \pmod{n}$  成立。当  $|p| \approx |q|$  时, 证明分解  $n$  的一个上界是  $n^{1/4}$ 。

提示: 应用 Pollard  $\lambda$  算法从  $g^{n+1} \pmod{n}$  中求出  $p+q$ ; 接着用  $p+q$  和  $pq$  分解  $n$ 。

6.9 设  $p$  为素数, 证明群  $\mathbb{Z}_p^*$  的生成元必为二次非剩余; 类似地, 设  $n$  是奇合数, 证明  $\mathbb{Z}_n^*$  中阶最大的元必为二次非剩余。

6.10 用欧拉准则验证模  $p$  的二次剩余比用计算勒让德符号方法验证慢了  $\log p$  倍, 为什么?

6.11 用习题 6.4 中计算的平方根分解 35。

6.12 证明  $\text{QR}_n$  是  $J_n(1)$  的一个子群, 后者是  $\mathbb{Z}_n^*$  的一个子群。

6.13 设  $n = pq$ , 其中  $p, q$  是不同的素数, 在什么条件下  $-1 \in \text{QR}_n$ ? 在什么条件下  $\left(\frac{-1}{n}\right) = -1$ ?

6.14 设  $n$  为 Blum 整数, 在集合  $\text{QR}_n$  上构造函数  $f(x) = x^2 \pmod{n}$  的逆。

提示: 对算法 6.3 的情形 1 运用中国剩余定理(算法 6.1)。

6.15 设  $n = pq$  为 Blum 整数, 满足  $\gcd(p-1, q-1) = 2$ 。证明  $J_n(1)$  是循环群。

提示: 分别找出  $\mathbb{Z}_p^*$  和  $\mathbb{Z}_q^*$  的生成元, 运用中国剩余定理求出  $\mathbb{Z}_n$  中的这个元素, 证明这个元素属于  $J_n(1)$ , 并且阶为  $\#J_n(1)$ 。



## 第三部分 基本的密码学技术

这一部分包括四章内容,介绍实现保密和数据完整性的大多数基本的密码技术。第7章介绍对称加密技术,第8章介绍非对称加密技术,第9章考虑当基本的和通用的非对称密码函数用于理想世界时(在这里数据是随机的),它们所具有的重要安全性质,最后第10章介绍实现数据完整性的基本技术。

由于这一部分将要介绍的基本密码算法和方案可以在许多的密码学教科书中找到,所以我们可以把它们看做是“教科书式密码”。在这一部分我们通过展示大量的攻击,揭示出这些“教科书式密码”算法和方案的各种各样的弱点,我们暂时并不给出补救这些弱点的方法,事实上,我们现在也给不出这些方法。然而,本书并不止限于介绍密码学的“教科书式密码”,非教科书式的加密算法和数据完整性机制也将在后面几章中介绍,它们中的大多数都是对“教科书式密码”相应部分增强的结果。

对于那些并不打算深入研究实用密码及其强安全性概念的读者,“教科书式密码”部分仍会就“教科书式密码”的普遍不安全性及早地给他们提出明确的警示。

## 第7章 加密——对称技术

### 7.1 引言

保密是密码学的核心,而加密是获得信息保密的实用工具。现代加密技术就是一些数学变换(算法),将消息看做是空间中的数字或代数元,然后在“有意义的消息”区和“不可理解的消息”区之间进行变换。把有意义区域中的消息和加密算法中的输入称为原文,而把加密算法不可理解的输出称为密文。如果我们忽视消息的可理解性,那么加密算法中的消息输入习惯上称为明文(码),它可能是可理解的,也可能是不可理解的。例如,明文消息可以是任意的一次性随机数消息或密文消息,在第2章讨论的某些协议中我们已经见到过那样的例子。因此,明文(码)和密文(码)是一对不同的概念:前者指输入到加密算法中的消息,而后者指从加密算法中输出的消息。

为了恢复信息,加密变换必须是可逆的,逆变换称为解密。通常,加密算法和解密算法都有密钥做参数,加密算法和解密算法再加上消息和密钥的形式描述就构成了一个密码系统或密码体制。

香农对密码体制所希望的性质有如下语义的刻画:密文消息空间是所有可能的消息空间,而原文(注意,按照上面第一段给出的习惯不是指明文)消息空间是消息空间中很小的一个区域,在这个区域中的消息具有某种相当简单的统计结构,也就是说,它们是有意义的;(好的)加密算法是一种混合变换,它将有意义的小区域中的有意义的消息相当均匀地分布到在整个消息空间中[266]。香农刻画这种混合特性如下:

$$\lim_{n \rightarrow \infty} \bigcup_n F^n R = \Omega \quad (7.1.1)$$

这里, $F$ 表示空间(消息空间)自身的一个映射(加密算法), $R$ 表示 $\Omega$ 中初始的小区域(原文区域)。香农对于加密的语义刻画说明了一个好的加密算法应该有这样一个混合变换性:它可以将空间中的一个初始小区域映射到整个空间。

尽管到今天,尤其是在公钥密码学出现之后,加密算法未必再是由空间到空间自身的映射(对于绝大多数的密码体制,单钥的或公钥的,仍是这样),香农将加密作为一种混合变换的语义刻画仍是非常有意义的。加密算法的语义安全的现代定义的实质含义是,密文在消息空间中的分布与同样空间中的均匀分布是不可区分的,这一定义将在14.3节中给出。

#### 7.1.1 本章概述

本章我们将介绍密码体制的概念、几个著名的对称密码体制和标准的运行模式。首先我们给出将在本书的其余各章节用到的密码体制的形式化语法定义(7.2节),然后我们介绍几个重要的古典密码(7.3节~7.4节),通过展示古典密码在现代密码和密码协议中的广泛作用,明确古典密码技术的重要性(7.5节)。在介绍完古典密码之后,描述两个重要的现代分组密码:数据加密标准(DES,7.6节)和高级加密标准(AES,7.7节),并且解释它们的设计策略,



我们还就 AES 对应用密码学产生的积极影响给出一个简短的讨论(7.7.5 节)。对称技术部分还将介绍各种标准的运行模式,以便于使用分组密码来获得概率加密(7.8 节)。最后,我们通过提出密钥信道建立的经典问题来结束对对称加密技术的介绍(7.9 节)。

## 7.2 定义

密码体制的语法定义如下。

**定义 7.1 密码体制** 密码体制构成如下:

- 明文消息空间  $\mathcal{M}$ : 某个字母表上的串集
- 密文消息空间  $\mathcal{C}$ : 可能的密文消息集
- 加密密钥空间  $\mathcal{K}$ : 可能的加密密钥集; 解密密钥空间  $\mathcal{K}'$ : 可能的解密密钥集
- 有效的密钥生成算法  $\mathcal{G}: \mathcal{N} \rightarrow \mathcal{K} \times \mathcal{K}'$
- 有效的加密算法  $\mathcal{E}: \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$
- 有效的解密算法  $\mathcal{D}: \mathcal{C} \times \mathcal{K}' \rightarrow \mathcal{M}$

对于整数  $1^\ell$ ,  $\mathcal{G}(1^\ell)$  输出长为  $\ell$  的密钥对  $(ke, kd) \in \mathcal{K} \times \mathcal{K}'$ 。

对于  $ke \in \mathcal{K}$  和  $m \in \mathcal{M}$ , 我们将加密变换表示为

$$c = \mathcal{E}_{ke}(m)$$

读做“ $c$  是  $m$  在密钥  $ke$  下的加密”; 将解密变换表示为

$$m = \mathcal{D}_{kd}(c)$$

读做“ $m$  是  $c$  在密钥  $kd$  下的解密”。对于所有的  $m \in \mathcal{M}$  和所有的  $ke \in \mathcal{K}$ , 一定存在  $kd \in \mathcal{K}'$ :

$$\mathcal{D}_{kd}(\mathcal{E}_{ke}(m)) = m \quad (7.2.1)$$

在本书的其余各章,除了那些文献上已经习惯使用不同记号的地方,我们将使用这个构造性的记号集来表示抽象的密码体制。图 7.1 给出了密码体制的图示。

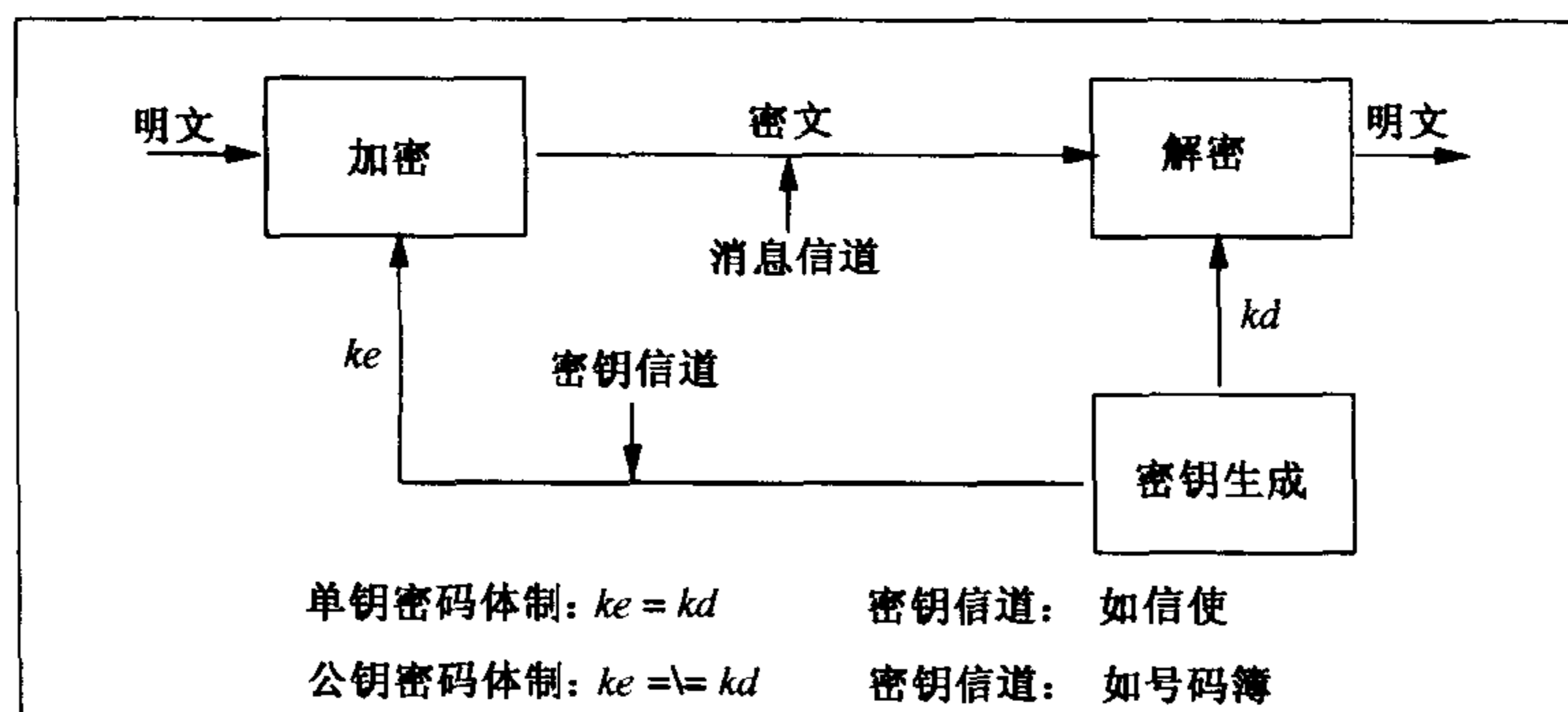


图 7.1 密码体制

定义 7.1 应用于既使用私钥又使用公钥(公钥密码体制将在下一章中介绍)的密码体制。在单钥密码体制中,加密和解密使用同样的密钥,加密消息的人必须和将要收到并解密已加密消息的人分享加密密钥。 $kd = ke$  的情况给了单钥密码体制另一个名字:对称密码体制。在

公钥密码体制中,加密和解密使用不同的密钥,对于每一个  $ke \in \mathcal{K}$ ,存在  $kd \in \mathcal{K}'$ ,这两个密钥是不同的并且互相匹配;加密密钥  $ke$  不必保密, $ke$  的拥有者可以使用相匹配的私钥  $kd$  来解密在  $ke$  下加密过的密文。 $kd \neq ke$  的情况给了公钥密码体制另一个名字:非对称密码体制。

由于要求加密算法的有效性,我们要考虑包括概率多项式时间算法在内的一些算法。因此,尽管抽象记号看起来是确定性的,它也有一个内在的随机变动性,所以输出密文是这个内在随机变动性的一个随机变量。同时注意到密钥生成算法  $\mathcal{G}$  的整数输入给出了输出加密/解密密钥的大小。既然密钥生成算法是有效的,有输入大小的多项式运行时间,输入整数值就应该使用单项表示(在 4.4.6.1 节中说明了理由)。

1883 年, Kerchoffs 列了一个设计密码要求必备的条件表(见[200])。在 Kerchoffs 列表中有一条已经发展为被广泛认可的约定,称为 **Kerchoffs 原理**:

知道算法和密钥的长度还可以获得已知的明文是现代密码分析的标准假设,既然敌手最终可以获得这些信息,那么评估密码强度时最好不要依赖于这些信息的保密性。

结合香农对密码体制的语义描述和 Kerchoffs 原理,我们可以对好的密码体制做如下总结:

- 算法  $\mathcal{E}$  和  $\mathcal{D}$  不包含秘密的成分或设计部分;
- $\mathcal{E}$  将有意义的消息相当均匀地分布在整个密文消息空间中;甚至可以由  $\mathcal{E}$  的某些随机的内部运算来获得随机的分布;
- 使用正确的密钥,  $\mathcal{E}$  和  $\mathcal{D}$  是实际有效的;
- 不使用正确的密钥,要由密文恢复出相应的明文是一个由密钥参数的大小惟一决定的困难问题,通常取长为  $s$  的密钥,使得解这个问题所要求计算资源的量级超过  $p(s)$ ,  $p$  是任意多项式。

我们应该注意,希望密码体制具有的这一览性质对于现代密码体制的应用来说已经不够了。通过对密码体制的研究,我们将会找到一些更为严格的要求。

## 7.3 代换密码

在代换密码中,加密算法  $\mathcal{E}_k(m)$  是一个代换函数,它将每一个  $m \in \mathcal{M}$  代换为相应的  $c \in \mathcal{C}$ 。代换函数的参数是密钥  $k$ 。解密算法  $\mathcal{D}_k(c)$  只是一个逆代换。通常,代换可由映射  $\pi: \mathcal{M} \mapsto \mathcal{C}$  给出,而逆代换恰是相应的逆映射  $\pi^{-1}: \mathcal{C} \mapsto \mathcal{M}$ 。

### 7.3.1 简单的代换密码

**例 7.1 简单的代换密码** 令  $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$ , 所包含元素表示为  $A = 0, B = 1, \dots, Z = 25$ 。将加密算法  $\mathcal{E}_k(m)$  定义为下面的  $\mathbb{Z}_{26}$  上的一个置换:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 21 & 12 & 25 & 17 & 24 & 23 & 19 & 15 & 22 & 13 & 18 & 3 & 9 \end{pmatrix}$$

$$\begin{pmatrix} 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 \\ 5 & 10 & 2 & 8 & 16 & 11 & 14 & 7 & 1 & 4 & 20 & 0 & 6 \end{pmatrix}$$

那么相应的解密算法 $\mathcal{D}_K(C)$ 为

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 24 & 21 & 15 & 11 & 22 & 13 & 25 & 20 & 16 & 12 & 14 & 18 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 \\ 9 & 19 & 7 & 17 & 3 & 10 & 6 & 23 & 0 & 8 & 5 & 4 & 2 \end{pmatrix}$$

明文消息

proceed meeting as agreed

加密为下面的密文消息(空间并不改变)

cqkzyyr jyyowft vl vtqyyr

□

在这个简单代换密码例子里,消息空间 $\mathcal{M}$ 和 $\mathcal{C}$ 都是字母表 $\mathbb{Z}_{26}$ 。换句话说,一个明文或密文消息是字母表中的单个字符。由于这个原因,明文消息串proceedmeetingasagreed并不是单个的消息,而是包含了22个消息。同样,密文消息串cqkzyyrjyyowftvltqyyr也包含22个消息。密码的密钥空间大小为 $26! > 4 \times 10^{26}$ ,与消息空间的大小相比是非常大的。然而,事实上这种密码是非常弱的:每一个明文字符被加密成惟一的密文字符。这一弱点致使这种密码对于称为频度分析的一种密码分析技术来说是相当脆弱的。频度分析揭示出一个事实,就是自然语言包含大量的冗余(回顾3.8节)。我们将在7.5节中进一步讨论简单代换密码的安全性。

历史上出现过几种特殊的简单代换密码,最简单且最著名的密码称为移位密码。在移位密码中, $\mathcal{K} = \mathcal{M} = \mathcal{C}$ ;令 $N = \#\mathcal{M}$ ,则加密和解密映射定义为

$$\begin{cases} \mathcal{E}_k(m) \leftarrow m + k \pmod{N} \\ \mathcal{D}_k(c) \leftarrow c - k \pmod{N} \end{cases} \quad (7.3.1)$$

其中 $m, c, k \in \mathbb{Z}_N$ 。当 $\mathcal{M}$ 为拉丁字母表的大写字母时,也就是 $\mathcal{M} = \mathbb{Z}_{26}$ ,移位密码也称为凯撒密码,这是因为Julius Caesar使用了该密码当 $k = 3$ 时的情形([94]的2.2节)。

由定理6.6(见6.2.2节)我们知道,如果 $\gcd(k, N) = 1$ ,那么对每个 $m < N$ 有:

$$km \pmod{N}$$

可取遍整个消息空间 $\mathbb{Z}_N$ 。因此,对于这样的 $k$ 和 $m, c < N$

$$\begin{cases} \mathcal{E}_k(m) \leftarrow km \pmod{N} \\ \mathcal{D}_k(c) \leftarrow k^{-1}c \pmod{N} \end{cases} \quad (7.3.2)$$

给出了一种简单代换密码。类似地,

$$k_1 m + k_2 \pmod{N}$$

也可以定义一种称为仿射密码的简单代换密码:

$$\begin{cases} \mathcal{E}_k(m) \leftarrow k_1 m + k_2 \pmod{N} \\ \mathcal{D}_k(c) \leftarrow k_1^{-1}(c - k_2) \pmod{N} \end{cases} \quad (7.3.3)$$

不难看出,利用 $\mathcal{K}$ 中密钥与 $\mathcal{M}$ 中消息之间的不同算术运算可以设计不同的简单代换密码,这些密码称为单表密码:对于一个给定的加密密钥,明文消息空间中的每一元素将被替换为密文消息空间中的惟一元素。因此,单表密码不能抵抗频度分析攻击。

然而,由于简单代换密码的简易性,它们已经被广泛应用于现代单钥加密算法中。我们将看到简单代换密码在数据加密标准(DES,7.6节)和高级加密标准(AES,7.7节)中所起到的核心作用。几个简单密码算法的结合可以产生一个安全的密码算法,这一点已经得到大家的认可,这就是简单密码仍被广泛应用的原因。简单代换密码在密码协议上也有广泛的应用,我们将在7.5节中阐明简单代换密码在密码协议中的应用,在本书的其余部分可以看到更多那样的例子。

### 7.3.2 多表密码

如果 $\mathcal{P}$ 中的明文消息元可以代换为 $\mathcal{C}$ 中的许多、可能是任意多的密文消息元,这种代换密码就称为多表密码。

由于维吉尼亚密码是多表密码中最知名的密码,所以我们将以它为例来说明多表密码。

维吉尼亚密码是基于串的代换密码:密钥是由多于一个的字符所组成的串。令 $m$ 为密钥长度,那么明文串被分为 $m$ 个字符的小段,也就是说,每一小段是 $m$ 个字符的串,可能的例外就是串的最后一小段不足 $m$ 个字符。加密算法的运算同于密钥串和明文串之间的移位密码,每次的明文串都使用重复的密钥串。解密同于移位密码的解密运算。

**例 7.2 维吉尼亚密码** 令密钥串为 gold。利用编码规则  $A=0, B=1, \dots, Z=25$ , 这个密钥串的数字表示是(6,14,11,3)。明文串

proceed meeting as agreed

的维吉尼亚加密运算如下,这种运算就是逐字符模 26 加:

15	17	14	2	4	4	3	12	4	4	19
6	14	11	3	6	14	11	3	6	14	11
<hr/>										
21	5	25	5	10	18	14	15	10	18	4
8	13	6	0	18	0	6	17	4	4	3
3	6	14	11	3	6	14	11	3	6	14
<hr/>										
11	19	20	11	21	6	20	2	7	10	17

因此密文串是

vfzfkso pkseltu lv guchkr

□

其他的著名的多表密码还包括书本密码(也称为 **Beale 密码**,即密钥串是已协商好的书中的原文)和 Hill 密码。有关这些代换密码的详细描述见[94]的 2.2 节或[286]的 1.1 节中的例子。

### 7.3.3 弗纳姆密码和一次一密

弗纳姆密码是最简单的密码体制之一。如果我们假定消息是长为 $n$ 的比特串

$$m = b_1 b_2 \cdots b_n \in \{0,1\}^n$$

那么密钥也是长为 $n$ 的比特串

$$k = k_1 k_2 \cdots k_n \in_U \{0,1\}^n$$

(这里注意到符号“ $\in_U$ ”均匀随机地选取 $k$ )。一次加密一比特,通过将每个消息比特和相应的密钥比特进行比特 XOR(异或)运算来得到密文串  $c = c_1 c_2 \cdots c_n$ 。

$$c_i = b_i \oplus k_i$$

$1 \leq i \leq n$ , 这里运算 $\oplus$ 定义为

$\oplus$	0	1
0	0	1
1	1	0

因为 $\oplus$ 是模2加,所以减法等于加法,因此解密与加密相同。

考虑 $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1\}^*$ , 则弗纳姆密码是代换密码的特例。如果密钥串只使用一次,那么弗纳姆密码满足代换密码的两个强安全性条件,这一点我们将在7.5节中说明,在那里我们还将讨论一次一密弗纳姆密码提供的保密性是在信息理论安全性的意义上的,或者说,是无条件的。理解这种安全性的一个简单方法如下:如果密钥 $k$ 等于 $c \oplus m$ (逐比特的),由于任意 $m$ 能够产生 $c$ ,所以密文消息串 $c$ 不能提供(给窃听者)关于明文消息串 $m$ 的任何信息。

一次一密弗纳姆密码也称为一次一密钥密码。原则上,只要加密密钥的使用满足我们将在7.5节列出的安全代换密码须满足的两个条件,那么任何代换密码都是一次一密密码。然而,习惯上,只有使用逐比特异或运算的密码才称为一次一密密码。

与其他的代换密码(例如使用模26加的移位密码)相比,逐位异或运算(模2加)在电子电路中更容易实现,因为这个原因,逐位异或运算被广泛应用在现代单钥加密算法的设计中。在两个重要的现代密码DES(见7.6节)和AES(见7.7节)中就使用了它。

加密中的一次一密钥类型也被广泛应用在密码学协议中,在7.5.1节中我们就将看到这样的一个协议。

## 7.4 换位密码

通过重新排列消息中元素的位置而不改变元素本身来变换一个消息的密码称为换位密码(也称为置换密码)。换位密码是古典密码中除代换密码外的重要一类,它广泛应用于现代分组密码的构造。

考虑明文消息中的元素是 $\mathbb{Z}_b$ 中的字符时的情形;令 $b$ 为一固定的正整数,它表示消息分组的大小;令 $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_b)^b$ ,最后令 $\mathcal{K}$ 是所有的置换,也就是 $(1, 2, \dots, b)$ 的所有重排。

那么因为 $\pi \in \mathcal{K}$ ,置换 $\pi = (\pi(1), \pi(2), \dots, \pi(b))$ 是一个密钥。对于明文分组 $(x_1, x_2, \dots, x_b) \in \mathcal{P}$ ,这个换位密码的加密算法是

$$e_\pi(x_1, x_2, \dots, x_b) = (x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(b)})$$

令 $\pi^{-1}$ 表示 $\pi$ 的逆即 $\pi^{-1}(\pi(i)) = i, i = 1, 2, \dots, b$ ,那么这个换位密码相应的解密算法是

$$d_\pi(y_1, y_2, \dots, y_b) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \dots, y_{\pi^{-1}(b)})$$

对于长度大于分组长度 $b$ 的消息,该消息可分成多个分组,然后逐分组重复同样的过程。

既然对于消息分组的长度 $b$ ,共有 $b!$ 种不同的密钥,因此一个明文消息分组能够变换加密为 $b!$ 种可能的密文。然而,由于字母本身并未改变,换位密码对于抗频度分析技术也是相当脆弱的。

**例 7.3 换位密码** 令  $b = 4, \pi = (\pi(1), \pi(2), \pi(3), \pi(4)) = (2, 4, 1, 3)$ , 那么明文消息

proceed meeting as agreed

首先分为 6 个分组, 每个分组 4 个字符:

proc eedm eeti ngas agre ed

然后可以变换-加密成下面的密文

rcpoemedei etgsnag earde

注意到明文的最后一个短分组 ed 实际上填充成了 ed<sub>□□</sub>, 然后加密成 d<sub>□</sub>e<sub>□</sub>, 再从密文分组中删掉补上的空格

解密密钥是

$$\pi^{-1} = (\pi(1)^{-1}, \pi(2)^{-1}, \pi(3)^{-1}, \pi(4)^{-1}) = (2^{-1}, 4^{-1}, 1^{-1}, 3^{-1})$$

最终的缩短密文分组 de 只包含两个字母说明了在相应的明文分组中没有字符与  $3^{-1}$  和  $4^{-1}$  的位置相匹配, 因此在解密过程正确执行以前, 应该将空格重新插入到缩短的密文分组中它们原来的位置上, 以便将分组恢复成添加空格的形式 d<sub>□</sub>e<sub>□</sub>。□

注意到对于最后的明文分组较短的情况(比如, 例 7.3 的情形), 由于添加的字符暴露了所用密钥的信息, 因此在密文消息中不要留下例如<sub>□</sub>这样的添加字符。

## 7.5 古典密码: 使用和安全性

首先我们要指出古典密码的两个基本工作原理: 代换和换位, 仍是构造现代对称加密算法的最重要的核心技术。我们将清楚地看到代换和换位密码在两个重要的现代对称加密算法 DES 和 AES 中的结合, 这些我们将在 7.6 节和 7.7 节中介绍。

考虑基于字符的代换密码。因为明文消息空间就是字母表, 每一个消息就是字母表中的一个字符, 加密就是逐字符地将每一明文字符代换为一个密文字符, 代换取决于密钥。在加密一个长字符串时, 如果密钥是固定的, 那么在明文消息中同一个字符将被加密成密文消息中一个固定的字符。

众所周知, 自然语言中的字符有稳定的频度(回顾 3.8 节), 自然语言中的字符频度分布知识为密码分析(由已知密文消息发现明文或加密密钥信息的技术)提供了线索。例 7.1 表明了这一情形, 该例中的字符 y 在密文消息中高频出现, 这表明一定有一个固定的字符在相应的明文消息中以相同的频率出现(事实上这个字符就是 e, 在英语中它是一个高频出现的字符)。简单代换密码不能隐藏基于自然语言的信息。基于字符频度研究的密码分析技术的详细内容可参阅密码学的任何一本标准教科书, 例如[94]中的 2.2 节或[200]中的 7.3.5 节。

多表密码和换位密码都比简单代换密码安全, 但是, 如果密钥很短而消息很长, 那么就有各种各样的密码分析技术能够攻破这样的密码。

然而, 如果密钥的使用满足了某些条件, 那么古典密码, 甚至是简单代换密码也可以是非常安全的。事实上, 在正确地使用了密钥以后, 简单代换密码可以广泛应用于密码体制和协议。



### 7.5.1 古典密码的使用

让我们来看一个被安全用于密码协议中的移位密码(也就是最简单的代换密码)的例子。给出这个例子之后,我们将总结出安全使用古典密码的两个重要条件。

假定我们有一个 $\mathbb{Z}_n$ 上的函数 $f(x)$ ,具有以下两条性质:

**单向:**对任意的 $x \in \mathbb{Z}_n$ ,  $f(x)$ 能够有效地计算(关于有效计算的含义,回顾4.4.6节),而对几乎所有的 $x \in \mathbb{Z}_n$ ,对于任意有效的算法 $A$ ,  $\text{Prob}[x \leftarrow A(y) \wedge f(x) = y]$ 与 $y$ 的大小相比是一个可忽略的量(关于可忽略量的含义,回顾4.6节);

**同态:**对于所有的 $x_1, x_2 \in \mathbb{Z}_n$ ,  $f(x_1 + x_2) = f(x_1) \cdot f(x_2)$ 。

有许多明显满足这两条性质的函数;我们以后将在本书中见到许多这样的函数。

利用这种函数,我们能够构造所谓的“零知识证明”协议,该协议能够让示证者(令其为 Alice)向验证者(令其为 Bob)证明她知道 $f(z)$  ( $z < n$ )的原像,但不会将原像透露给后者。这可以用一个使用移位密码的简单协议来实现。该协议在协议7.1中给出。

#### 协议 7.1 利用移位密码的零知识协议

共同输入 i)  $f():\mathbb{Z}_n$  上的一个单向同态函数;  
ii) 对于某个  $z \in \mathbb{Z}_n$ ,  $X = f(z)$ 。

Alice 的输入  $z < n$ 。( \* 示证者的私人输入 \* )

给 Bob 的输出 Alice 知道  $z \in \mathbb{Z}_n$ , 满足  $X = f(z)$ 。

重复下面的步骤  $m$  次:

1. Alice 选取  $k \in_U \mathbb{Z}_n$ , 计算  $\text{Commit} \leftarrow f(k)$  并发送给 Bob;
2. Bob 选取  $\text{Challenge} \in_U \{0, 1\}$  并发送给 Alice;
3. Alice 计算  $\text{Response} \leftarrow \begin{cases} k & \text{如果 Challenge} = 0 \\ k + z(\text{mod } n) & \text{如果 Challenge} = 1 \end{cases}$ , 然后将 Response 发送给 Bob; ( \* 当  $\text{Challenge} = 1$  时, Response 是在一次性密钥  $k$  下  $z$  的移位密码加密的密文输出, 见式(7.3.1) \* )
4. Bob 验证  $f(\text{Response}) = \begin{cases} \text{Commit} & \text{如果 Challenge} = 0 \\ \text{Commit} \cdot X & \text{如果 Challenge} = 1 \end{cases}$ , 如果任何验证步骤出现错误, 他将拒绝并中断运行; Bob 接受。

协议 7.1 是非常有用的。在实用中, 值  $X = f(z)$  可以成为 Alice 向服务器证明她的身份或权利的密码凭证。只有 Alice 知道怎样使用该凭证, 因为事实上只有 Alice 知道原像  $z$ , 所以也只有她才能使用这个凭证。这一协议证明了 Alice 怎样使用她的凭证而不让验证者 Bob 知道关于原像  $z$  的任何信息。

在第 18 章, 当我们研究零知识证明协议的时候, 我们将广泛地使用该协议和它的几个变形。目前, 我们所关注的是对于隐藏 Alice 的私人信息  $z$ , 这个协议所能提供的保密性。

### 7.5.2 古典密码的安全性

现在让我们看看协议 7.1 给出的移位密码加密的保密性。我们说保密性是极好的。就是



说,在运行完该协议以后,Bob 得到的关于  $z \in \mathbb{Z}_n$  的信息决不超过他已经从公共输入  $f(z)$  得到的信息(公共输入只提供先验信息)。

我们应该注意到移位密码加密

$$\text{Response} = z + k \pmod{n}$$

形成  $\mathbb{Z}_n$  上的一个置换。 $k \in {}_U \mathbb{Z}_n = \mathcal{K} = \mathcal{M}$ , 由于置换将均匀分布映射为均匀分布, 所以这个置换使得  $\text{Response} \in {}_U \mathbb{Z}_n$ , 这意味着对一个给定的密文  $\text{Response}$ ,  $\mathbb{Z}_n$  上的任何密钥都能够等可能地用来生成  $\text{Response}$  (概率空间是密钥空间和消息空间), 这等价于说任何  $x \in \mathbb{Z}_n$  等可能地被加密成  $\text{Response}$ 。所以明文  $z$  独立于密文  $\text{Response}$ , 或者说密文不泄漏关于明文的任何信息。

如果一个密码的明文分布与密文分布是独立的, 那么我们说这个密码在信息理论安全的意义上是安全的。与第 4 章我们建立的复杂性理论意义上的安全性相比, 信息理论意义上的安全性是无条件的并且能够抵抗任何方法的密码分析。在协议 7.1 中, 这种意义的安全意味着协议的运行不会提供给 Bob 任何关于 Alice 私人输入  $z$  的知识, 而只能确认 Alice 有正确的私人输入。

基于信息理论密码安全性的概念是由香农发展起来的[266]。按照香农理论, 我们可以总结出古典密码安全使用的两个条件:

**古典密码安全使用的条件**

- i)  $\#\mathcal{K} \geq \#\mathcal{M}$ ;
- ii)  $k \in {}_U \mathcal{K}$ , 且每次加密只使用一次。

所以如果使用古典密码(无论是基于字符或是基于串的简单代换密码、多表密码还是弗纳姆密码)加密长为  $\ell$  的消息串, 那么为了使加密是安全的, 密钥串的长度应该至少是  $\ell$ , 并且密钥串应该只使用一次。这个要求对于那些需要加密大量信息的应用来说不是很实际, 然而无疑对于加密少量数据是实用的, 例如一次性随机数(见 2.6.4 节)或一个会话密钥(见 2.5 节)。协议 7.1 就是这样的一个例子。

在本书的其余部分, 我们将看到许多密码体制和协议, 它们都应用了代换密码的各种形式, 例如移位密码(如协议 7.1)、乘积密码[在式(7.3.2)中定义]、仿射密码[在式(7.3.3)中定义]和一般置换形式下的代换密码(如例 7.1)。绝大多数这样的应用都满足古典密码安全使用的两个条件。

## 7.6 数据加密标准(DES)

毫无疑问, 数据加密标准(DES)中的算法是第一个并且也是最重要的现代对称加密算法[213]。1977 年 1 月, 美国国家标准局公布了 DES, 它是用于非保密数据(与国家安全无关的信息)的算法。该算法在世界范围内已经得到了广泛的应用, 一个主要的例子就是银行将它用于资金转账安全。本来该标准被批准使用五年, 由于它经受住了时间的考验, 随后又批准了三个五年使用期。

### 7.6.1 介绍 DES

DES 是分组密码, 其中的消息被分成定长的数据分组, 每一分组称为  $M$  或  $C$  中的一个消息。

在 DES 中,有  $\mathcal{M} = \mathcal{C} = \{0,1\}^{64}$ ,  $\mathcal{K} = \{0,1\}^{56}$ ,也就是 DES 加密和解密算法输入 64 比特明文或密文消息和 56 比特密钥,输出 64 比特密文或明文消息。

DES 的运算可描述为如下三步:

1. 对输入分组进行固定的“初始置换”IP,我们可以将这个初始置换写为

$$(L_0, R_0) \leftarrow \text{IP}(\text{输入分组}) \quad (7.6.1)$$

这里  $L_0$  和  $R_0$  称为“(左,右)半分组”,都是 32 比特的分组。注意到 IP 是固定的函数(也就是说,输入密钥不是它的参数),是公开的,因此这个初始置换无明显的密码意义。

2. 将下面的运算迭代 16 轮( $i = 1, 2, \dots, 16$ ):

$$L_i \leftarrow R_{i-1} \quad (7.6.2)$$

$$R_i \leftarrow L_{i-1} \oplus f(R_{i-1}, k_i) \quad (7.6.3)$$

这里  $k_i$  称为“轮密钥”,它是 56 比特输入密钥的一个 48 比特的子串, $f$  称为“S 盒函数”(“S”表示代换,我们将在 7.6.2 节中给出这个函数的一个简单描述),是一个代换密码(见 7.3 节)。这个运算的特点是交换两半分组,就是说,一轮的左半分组输入是上一轮的右半分组输出。交换运算是一个简单的换位密码(见 7.4 节),目的是获得很大程度的“信息扩散”,本质上就是获得式(7.1.1)中香农提出的模型的混合特性。从我们的讨论可以看出,DES 的这一步是代换密码和换位密码的结合。

3. 将 16 轮迭代后得到的结果  $(L_{16}, R_{16})$  输入到 IP 的逆置换来消除初始置换的影响。这一步的输出就是 DES 算法的输出,我们将最后一步写为

$$\text{输出分组} \leftarrow \text{IP}^{-1}(R_{16}, L_{16}) \quad (7.6.4)$$

请特别注意  $\text{IP}^{-1}$  的输入:在输入  $\text{IP}^{-1}$  以前,16 轮迭代输出的两个半分组又进行了一次交换。

加密和解密算法都用这三个步骤,仅有的不同就是如果加密算法中使用的轮密钥是  $k_1, k_2, \dots, k_{16}$ ,那么解密算法中使用的轮密钥就应当是  $k_{16}, k_{15}, \dots, k_1$ 。这种排列轮密钥的方法称为“密钥表”,可以记为

$$(k'_1, k'_2, \dots, k'_{16}) = (k_{16}, k_{15}, \dots, k_1) \quad (7.6.5)$$

**例 7.4** 在加密密钥  $k$  下,将明文消息  $m$  加密为密文消息  $c$ 。让我们通过 DES 算法来确认解密函数的正确运行,也就是在  $k$  下, $c$  的解密将输出  $m$ 。

解密算法首先输入密文  $c$  作为“输入分组”。由式(7.6.1)我们有

$$(L'_0, R'_0) \leftarrow \text{IP}(c)$$

但是因为  $c$  实际上是加密算法中最后一步的“输出分组”,由式(7.6.4)我们有

$$(L'_0, R'_0) = (R_{16}, L_{16}) \quad (7.6.6)$$

在第一轮中,由式(7.6.2)、式(7.6.3)和式(7.6.6),我们有

$$L'_1 \leftarrow R'_0 = L_{16}$$

$$R'_1 \leftarrow L'_0 \oplus f(R'_0, k'_1) = R_{16} \oplus f(L_{16}, k'_1)$$

在这两个式子的右边,由式(7.6.2), $L_{16}$ 应该用 $R_{15}$ 代替,由式(7.6.3), $R_{16}$ 应该用 $L_{15} \oplus f(R_{15}, k_{16})$ 代替,根据密钥表(7.6.5), $k'_1 = k_{16}$ 。因此,上面两个式子实际上是下面的两个:

$$L'_1 \leftarrow R_{15}$$

$$R'_1 \leftarrow [L_{15} \oplus f(R_{15}, k_{16})] \oplus f(R_{15}, k_{16}) = L_{15}$$

所以,在第一轮解密以后,我们得到

$$(L'_1, R'_1) = (R_{15}, L_{15})$$

因此,在第二轮开始,两个半分组是 $(R_{15}, L_{15})$ 。

在随后的15轮中,使用同样的验证,我们将获得

$$(L'_2, R'_2) = (R_{14}, L_{14}), \dots, (L'_{16}, R'_{16}) = (R_0, L_0)$$

从16轮迭代得到的两个最后的半分组 $(L'_{16}, R'_{16})$ 被交换为 $(L'_{16}, R'_{16}) = (L_0, R_0)$ ,然后输入到 $IP^{-1}$ [注意到式(7.6.4)中另外一次的交换]来消除 $IP$ 在式(7.6.1)中的影响。解密函数的输出确实就是最初的明文分组 $m$ 。□

我们已经证明了DES加密和解密算法确实使得式(7.2.1)对于所有的 $m \in \mathcal{M}$ 和 $k \in \mathcal{K}$ 都成立,很明显这些算法的运行与“S盒函数”的内部细节及密钥表函数无关。

使用式(7.6.2)和式(7.6.3)以交换的方式处理两个半分组的DES迭代称为**Feistel密码**。图7.2给出了一轮Feistel密码的交换结构。最初是由Feistel提出了这个密码[108]。像我们以前提到的那样,交换特性目的是获得

一个比较大程度的数据扩散。Feistel密码在公钥密码学中也有重要的应用:称为**最佳非对称加密填充(OAEP)**的结构本质上是一个二轮的Feistel密码。我们将在15.2节中讨论OAEP。

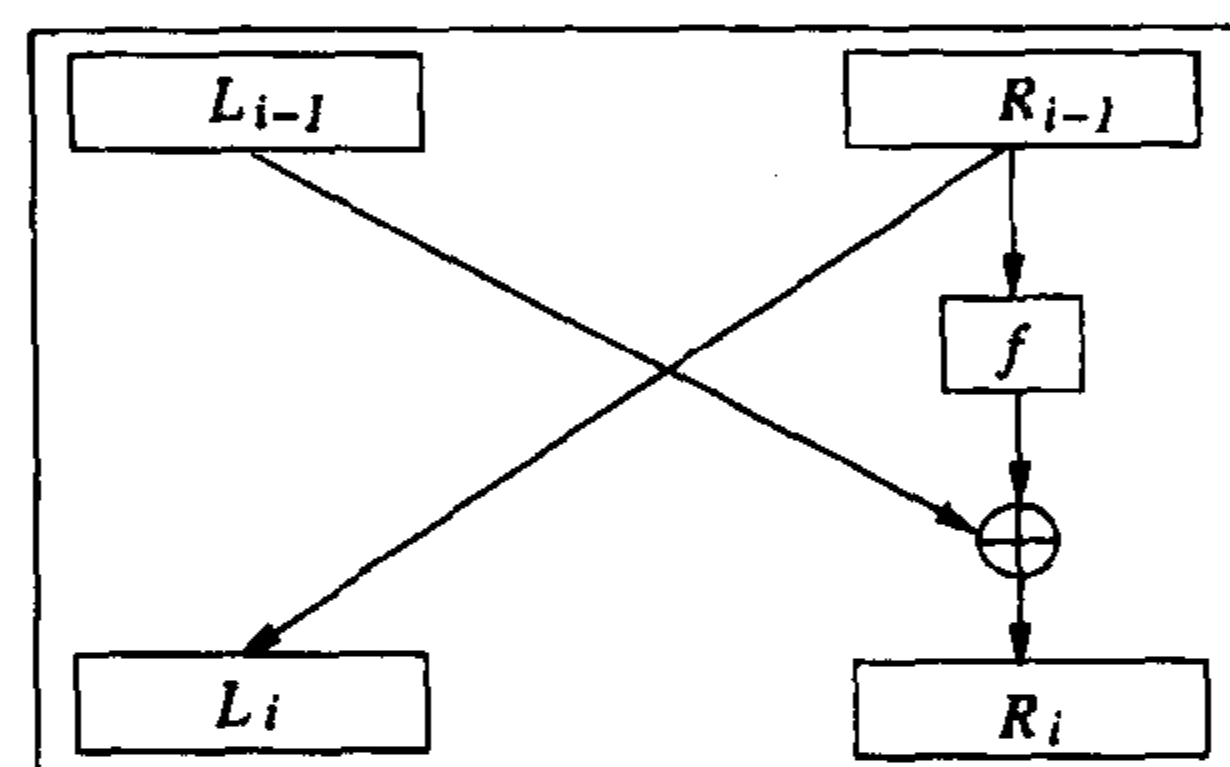


图 7.2 Feistel 密码体制(一轮)

## 7.6.2 DES 的核心作用:消息的随机非线性分布

DES 的核心部分是在“S盒函数” $f$ 里。正是在这里DES实现了明文消息在密文消息空间上的随机非线性分布。

在第 $i$ 轮, $f(R_{i-1}, k_i)$ 做下面的两个子运算:

- i) 通过逐比特异或运算,将轮密钥 $k_i$ 与半分组 $R_{i-1}$ 相加;这提供了消息分布中所需要的随机性;
- ii) 在包含八个“代换盒”(S盒)的固定置换下代换(i)的结果,每一个S盒是一个非线性置换函数;这就提供了消息分布中所需的非线性性。

S盒的非线性性对DES的安全是非常重要的。我们注意到代换密码(例如,有随机密钥的例7.1)在一般情况下是非线性的,而移位密码和仿射密码是线性中的子类。与一般的情况相比,这些线性子类不仅极大地减小了密钥空间的大小,而且也导致了生成的密文对于差分分析(DC, differential cryptanalysis)技术是脆弱的[34]。DC通过利用两个明文消息间的线性差分和

两个密文消息间的线性差分来攻击密码。让我们以仿射密码(7.3.3)为例来看看这样的攻击。假定 Malice(攻击者)以某种方式知道了差分  $m - m'$ , 但是他既不知道  $m$  也不知道  $m'$ 。给定相应的密文  $c = k_1 m + m_2 \pmod{N}$ ,  $c' = k_1 m' + k_2 \pmod{N}$ , Malice 可以计算

$$k_1 = (c - c') / (m - m') \pmod{N}$$

有了  $k_1$ , Malice 进一步找到  $k_2$  就变得容易多了。例如, 如果 Malice 有一个已知的明文-密文对, 他就能找到  $k_2$ 。在 1990 年发现了 DC 以后, 对于许多已知的分组密码的攻击, DC 已经被证明是非常有效的, 然而它攻击 DES 并不是非常成功。这就表明 DES 的设计者早在 15 年前通过 S 盒的非线性设计就已采取了预防 DC 的措施。

DES 的一个有趣的特点(事实上还有 Feistel 密码)就是函数  $f(R_{i-1}, k_i)$  中的 S 盒不必是可逆的。在例 7.4 中对于任意的  $f(R_{i-1}, k_i)$  都可运行加密和解密就证明了这一点。该特点节约了 DES 硬件实现的空间。

我们将省略对 S 盒的内部细节、密钥表函数和初始置换函数的描述。这些细节超出了本书的范围, 有兴趣的读者可在[94]的 2.6.2 节中找到这些细节。

### 7.6.3 DES 的安全性

在 DES 作为加密标准提出之后不久就开始争论 DES 的安全性。详细的讨论和历史描述可以在各种密码学教课书中找到, 如[281]中的第 2 节, [286]中的 3.3 节和[200]中的 7.4.3 节。后来, 越来越清楚的是这些讨论找到了 DES 的一个主要缺点: DES 的密钥长度较短。这被认为是 DES 仅有的最严重的弱点。针对这个弱点的攻击包括穷举测试密钥, 就是利用一个已知的明文和密文消息对, 直到找到正确的密钥。这就是所谓的强力或穷举密钥搜索攻击。

然而, 我们不能将强力密钥搜索攻击看做是一种真正的攻击, 这是因为密码设计者不仅已经预见了它, 而且希望这是对手仅有的工具。因此, 假设有 20 世纪 70 年代的计算技术, 那么 DES 是一种十分成功的密码。

克服短密钥缺陷的一个解决办法是使用不同的密钥, 多次运行 DES 算法。这样的方案称为加密-解密-加密-三重 DES 方案[292]。这个方案中的加密记为

$$c \leftarrow \mathcal{E}_{k_1}(\mathcal{D}_{k_2}(\mathcal{E}_{k_1}(m)))$$

解密记为

$$m \leftarrow \mathcal{D}_{k_1}(\mathcal{E}_{k_2}(\mathcal{D}_{k_1}(c)))$$

除了能够达到扩大密钥空间的效果, 如果使用, 这个方案也很容易与单钥 DES 兼容。三重 DES 也可以使用三个不同的密钥, 但这时它与单钥 DES 不兼容。

DES 的短密钥弱点在 20 世纪 90 年代变得明显了。1993 年, Wiener 认为花费 1 000 000 美元可以造一个特殊用途的 VLSI DES 密钥搜索机, 给定一个明文-密文消息对, 预计这台机器将在 3.5 个小时之内找到密钥[301]。1998 年 7 月 15 日, 密码学研究协会、高级无线技术协会和电子前沿基金会(EFF, Electronic Frontier Foundation)联合宣布了破纪录的 DES 密钥搜索攻击: 他们花了不到 250 000 美元构造了一个称为 DES 解密高手(也称为 Deep Crack)的密钥搜索机, 搜索了 56 个小时后成功地找到了 RSA 的 DES 挑战密钥[111]。这个结果表明 20 世纪 90 年代后期的计算技术对于一个安全的单钥密码来说, 使用 56 比特的密钥太短了。

## 7.7 高级加密标准(AES)

1997年1月2日,美国国家标准和技术协会(NIST)宣布征集一个新的对称密钥分组密码算法作为取代DES的新的加密标准。这个新的算法将被命名为高级加密标准(AES)。与DES的封闭设计过程不同,在1997年9月12日,正式地公开征集AES算法,规定了AES要详细说明一个非保密的、公开的对称密钥加密算法;算法必须支持(至少)128比特的分组长度,128、192和256比特的密钥长度,强度应该相当于三重DES,但是应该比三重DES更有效。此外,如果算法被选中的话,在世界范围内它必须是<sup>①</sup>可以免费获得的。

1998年8月20日,NIST公布了15个AES候选算法,这些算法由遍布世界的密码团体的成员提交。公众对这15个算法的评论被当做这些算法的初始评论(公众的初始评论期也称为第一轮),第一轮1999年4月15日截止。根据收到的分析和评论,NIST从15个算法中选出5个算法,这5个参加决赛的候选算法是MARS[63]、RC6[249]、Rijndael[87]、Serpent[15]和Twofish[257]。这些参加决赛的算法在又一次更深入的评论期(第二轮)得到进一步的分析。在第二轮中,要征询对候选算法的各方面的评论和分析,这些方面包括但不限于下面的方面:密码分析、知识产权、所有AES决赛候选算法的剖析、综合评价及有关实现问题。在2000年5月15日第二轮公众分析期结束以后,NIST研究了所有可得到的信息,以便于为AES作出选择。2000年10月2日,NIST宣布它已经选中了Rijndael来建议作为AES。

Rijndael是由两个比利时密码学家Daemen和Rijmen共同设计的。

### 7.7.1 Rijndael 密码概述

Rijndael是具有分组长度和密钥长度均可变的分组密码。密钥长度和分组长度可以独立地指定为128比特、192比特或256比特。为简化起见,我们只讨论最小,即密钥长度128比特、分组长度128比特时的情形。我们所限定的描述无损于Rijndael密码工作原理的一般性。

在这种情况下,128比特的消息(明文,密文)分组被分成16个字节(一个字节是8比特,所以有 $128 = 16 \times 8$ 比特),记为:

$$\text{输入分组} = m_0, m_1, \dots, m_{15}$$

密钥分组也是这样:

$$\text{输入密钥} = k_0, k_1, \dots, k_{15}$$

内部数据结构的表示是一个 $4 \times 4$ 矩阵:

$$\begin{aligned} \text{输入分组} &= \begin{pmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{pmatrix} \\ \text{输入密钥} &= \begin{pmatrix} k_0 & k_4 & k_8 & k_{12} \\ k_1 & k_5 & k_9 & k_{13} \\ k_2 & k_6 & k_{10} & k_{14} \\ k_3 & k_7 & k_{11} & k_{15} \end{pmatrix} \end{aligned}$$

同 DES(以及最现代的对称密钥分组密码)一样, Rijndael 算法也是由基本的变换单位——“轮”多次迭代而成。在消息分组长度和密钥分组均为 128 比特的最小情况, 轮数是 10。当消息长度和密钥长度变大的时候, 轮数也应该相应增加, 这在[221]的图 5 中给出。

Rijndael 中的轮变换记为

$$\text{Round}(\text{State}, \text{RoundKey})$$

这里 *State* 是轮消息矩阵, 既被看做是输入, 也被看做是输出; *RoundKey* 是轮密钥矩阵, 它是由输入密钥通过密钥表导出的。一轮的完成将导致 *State* 的元素改变值(也就是改变它的状态)。对于加密(对应地, 解密), 输入到第一轮中的 *State* 就是明文(对应地, 密文)消息矩阵输入分组, 而最后一轮中输出的 *State* 就是密文(对应地, 明文)消息矩阵。

轮(除了最后一轮)变换由四个不同的变换组成, 这些变换是将要介绍的内部函数:

```
Round( State, RoundKey ) {
    SubBytes( State );
    ShiftRows( State );
    MixColumns( State );
    AddRoundKey( State, RoundKey );
}
```

最后一轮有点不同, 记为

$$\text{FinalRound}(\text{State}, \text{RoundKey})$$

它等于不使用 MixColumns 函数的 Round( *State*, *RoundKey* )。这类似于 DES 中最后一轮的情形, 就是在输出的半数据分组之间再做一次交换。

轮变换是可逆的, 以便于解密。相应的逆轮变换分别记为

$$\text{Round}^{-1}(\text{State}, \text{RoundKey}) \text{ 和 } \text{FinalRound}^{-1}(\text{State}, \text{RoundKey})$$

下面我们将看到四个内部函数都是可逆的。

### 7.7.2 Rijndael 密码的内部函数

现在让我们描述 Rijndael 密码的四个内部函数。因为每一个内部函数都是可逆的, 为了实现 Rijndael 的解密, 我们只需要在相反的方向使用它们各自的逆就可以了, 因此我们仅就加密方向来描述这些函数。

Rijndael 密码的内部函数是在有限域上实现的,  $\mathbb{F}_2$  上的所有多项式模不可约多项式

$$f(x) = x^8 + x^4 + x^3 + x + 1$$

就得到了这个域。明确地说, Rijndael 密码所用的域是  $\mathbb{F}_2[x]_f$ , 这个域中的元素就是  $\mathbb{F}_2$  上次数小于 8 的多项式, 运算是模  $f(x)$  运算。我们把这个域称为“Rijndael 域”。由于同构关系, 我们经常用  $\mathbb{F}_{2^8}$  来表示这个域, 这个域中有  $2^8 = 256$  个元素。

实际上, 我们在第 5 章例 5.17、例 5.18 和例 5.19 中已经讨论了 Rijndael 域, 并且演示了下面的运算:

- 整数字节和域元素之间的一一映射(例 5.17)
- 两个域元素之间的加法(例 5.18)
- 两个域元素之间的乘法(例 5.19)



我们在那里的讨论现在能够帮助我们描述 Rijndael 的内部函数。

首先如我们已经描述的,在 Rijndael 密码中一个消息分组(一个状态)和一个密钥分组被分成字节。由例 5.17 中所描述的简单的一一映射表,这些字节可以看成是域元素并由我们现在要描述的几个 Rijndael 内部函数所作用。

### 7.7.2.1 内部函数 SubBytes( State)

这个函数为 *State* 的每一字节(也就是  $x$ )提供了一个非线性代换,任一非零字节  $x \in (\mathbb{F}_2^8)^*$  被下面的变换所代换:

$$y = Ax^{-1} + b \quad (7.7.1)$$

这里

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \text{ 和 } b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

如果  $x$  是零字节,那么  $y = b$  就是 SubBytes 变换的结果。

我们应该注意到在式(7.7.1)中变换的非线性仅仅来自于逆  $x^{-1}$ ,如果这个变换直接作用于  $x$ ,那么在式(7.7.1)中的仿射方程将绝对是线性的!

因为  $8 \times 8$  常数矩阵  $A$  是可逆的(也就是说,它的行在  $\mathbb{F}_2^8$  中是线形无关的),所以在式(7.7.1)中的变换是可逆的,因此函数 SubBytes( State)是可逆的。

### 7.7.2.2 内部函数 ShiftRows( State)

这个函数在 *State* 的每行上运算。对于 128 比特分组长度的情形,它就是下面的变换:

$$\begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix} \rightarrow \begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,1} & s_{1,2} & s_{1,3} & s_{1,0} \\ s_{2,2} & s_{2,3} & s_{2,0} & s_{2,1} \\ s_{3,3} & s_{3,0} & s_{3,1} & s_{3,2} \end{pmatrix} \quad (7.7.2)$$

这个运算实际上是一个换位密码(见 7.4 节),它只是重排了元素的位置而不改变元素本身:对于在第  $i$  ( $i = 0, 1, 2, 3$ ) 行的元素,位置重排就是“循环向右移动” $4 - i$  个位置。

既然换位密码仅仅重排行元素的位置,那么这个变换当然是可逆的。

### 7.7.2.3 内部函数 MixColumns( State)

这个函数在 *State* 的每列上作用,所以对于式(7.7.2)中右边的矩阵的四列 *State*, MixColumns( State)迭代四次。下面只描述了对一列的作用,一次迭代的输出仍是一列。



首先,令

$$\begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix}$$

是式(7.7.2)中右边矩阵中的一列,注意到为了表述清楚我们已经省略了列数。

把这一列表示为3次多项式:

$$s(x) = s_3 x^3 + s_2 x^2 + s_1 x + s_0$$

注意到因为  $s(x)$  的系数是字节,也就是说是在  $\mathbb{F}_2^8$  中的元素,所以这个多项式是在  $\mathbb{F}_2^8$  上的,因此不是 Rijndael 中的元素。

列  $s(x)$  上的运算定义为将这个多项式乘以一个固定的3次多项式  $c(x)$ ,然后模  $x^4 + 1$ :

$$c(x) \cdot s(x) \pmod{x^4 + 1} \quad (7.7.3)$$

这里固定的多项式  $c(x)$  是

$$c(x) = c_3 x^3 + c_2 x^2 + c_1 x + c_0 = \text{'03'} x^3 + \text{'01'} x^2 + \text{'01'} x + \text{'02'}$$

$c(x)$  的系数也是  $\mathbb{F}_2^8$  中的元素(以十六进制表示字节或域元素)。

我们应该注意到式(7.7.3)中的乘法不是 Rijndael 域中的运算: $c(x)$  和  $s(x)$  甚至不是 Rijndael 域中的元素。而且因为  $x^4 + 1$  在  $\mathbb{F}_2$  ( $x^4 + 1 = (x + 1)^4$ ) 上可约,在式(7.7.3)中的乘法甚至不是任何域中的运算(见 5.4.2.2 节中的定理 5.5)。进行乘法模一个4次多项式的仅有理由就是为了使运算输出一个3次多项式,也就是说,为了获得一个从一列(3次多项式)到另一列(3次多项式)的变换,这个变换可以看做是使用已知密钥的一个多表代换(乘积)密码。

读者可以使用例 5.15 中的长除法来验证下面在  $\mathbb{F}_2$  上计算的方程(注意到在这个环中,减法与加法等同):

$$x^i \pmod{x^4 + 1} = x^{i \pmod{4}}$$

因此,在式(7.7.3)的乘积中, $x^i$  ( $i = 0, 1, 2, 3$ ) 的系数一定是满足  $j + k = i \pmod{4}$  的  $c_j s_k$  的和(这里  $j, k = 0, 1, 2, 3$ )。例如,在乘积中  $x^2$  的系数是

$$c_2 s_0 + c_1 s_1 + c_0 s_2 + c_3 s_3$$

因为乘法和加法都在  $\mathbb{F}_2^8$  中,所以很容易验证式(7.7.3)中的多项式乘法可由下面的线性代数式给出:

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} = \begin{pmatrix} \text{'02'} & \text{'03'} & \text{'01'} & \text{'01'} \\ \text{'01'} & \text{'02'} & \text{'03'} & \text{'01'} \\ \text{'01'} & \text{'01'} & \text{'02'} & \text{'03'} \\ \text{'03'} & \text{'01'} & \text{'01'} & \text{'02'} \end{pmatrix} \begin{pmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{pmatrix} \quad (7.7.4)$$

进一步注意到,因为在  $\mathbb{F}_2$  上  $c(x)$  与  $x^4 + 1$  是互素的,所以在  $\mathbb{F}_2[x]$  中逆  $c(x)^{-1} \pmod{x^4 + 1}$  是存在的,这等价于说矩阵,因而也就是式(7.7.4)中的变换是可逆的。

#### 7.7.2.4 内部函数 $\text{AddRoundKey}(State, RoundKey)$

这个函数仅仅是逐字节、逐比特地将  $RoundKey$  中的元素与  $State$  中的元素相加,这里的“加”是  $\mathbb{F}_2$  中的加法(也就是逐比特异或),是平凡可逆的;逆就是自身相“加”。

$RoundKey$  比特已经被列表,也就是说,不同轮的密钥比特是不同的,它们由使用一个固定的(非秘密的)“密钥表”方案的密钥导出。有关“密钥”表的细节请参阅[221]的图 12。

到现在为止,我们已经完成了 Rijndael 内部函数的描述,因此也完成了加密运算的描述。

#### 7.7.2.5 解密运算

如我们已经看到的,四个内部函数都是可逆的,因此解密仅仅是在相反的方向反演加密,也就是说,运行

$$\begin{aligned} & \text{AddRoundKey}(State, RoundKey)^{-1}; \\ & \text{MixColumns}(State)^{-1}; \\ & \text{ShiftRows}(State)^{-1}; \\ & \text{SubBytes}(State)^{-1}. \end{aligned}$$

我们应当注意到,与 Feistel 密码不同,Feistel 密码的加密和解密可以使用同样的电路(硬件)和代码(软件),而 Rijndael 密码的加密和解密必须分别使用不同的电路和代码。

### 7.7.3 Rijndael 内部函数的功能小结

在我们结束对 Rijndael 密码描述之前,让我们对四个内部函数的功能给出一个小结。

- SubBytes 的目的是为了得到一个非线性的代换密码。如我们在 7.6.2 节中讨论的那样,对于分组密码抗差分分析来说,非线性是一个重要的性质。
- ShiftRows 和 MixColumns 的目的是获得明文消息分组在不同位置上的字节的混合。有代表性的比如,由于在自然语言和商业数据中包含的高冗余导致的明文消息在消息空间有一个低熵分布(也就是说,典型的明文集中在整个消息空间中的一个较小的子空间中),而消息分组中不同位置上的字节的混合导致了消息在整个消息空间中更广的分布。这本质上就是 7.1.1 节中香农提出的混合特性。
- AddRoundKey 给出了消息分布所需的秘密随机性。

这些函数重复多次(对于 128 比特密钥和数据长度的情形,至少重复 10 次)以后,就构成了 Rijndael 密码。

#### 7.7.4 快速而安全的实现

我们已经看到 Rijndael 内部函数是非常简单的,它在很小的代数空间上运算,因此可以高效地实现这些内部函数。从我们对 Rijndael 内部函数的描述可以看出,只有 SubBytes 和 MixColumns 有非平凡的代数运算,因此值得考虑它们的快速实现。

首先,在 SubBytes 中,利用“查表法”能够有效计算  $x^{-1}$ :可以一次建立一个有  $2^8 = 256$  个字节对的小表,长期使用(也就是说,这个表可以“固化”嵌入硬件或者是软件中实现)。在这个由对组成的表中,零字节与零字节对应,表中的其余 255 项是  $(x, x^{-1})$  对的 255 种情况。“查表

法”不仅是非常有效的,还能抗定时分析攻击,这种攻击根据观察不同数据的运算时间差异,就能够暗示出一个运算是在比特 0 上进行还是在比特 1 上进行(见 12.5.4 节)。

因为式(7.7.1)中的矩阵  $A$  和向量  $b$  是常量,“查表法”实际上可以完全包括式(7.7.1)的整个变换,也就是说,256 项的表是由对  $(x, y)$  组成,其中  $x, y \in \mathbb{F}_2^8$ ,而  $(0, b)$  是  $(x, y)$  的特殊情况。

显然,只要使用逆元表就可以得到逆元,因此,SubBytes 可以用两个小表来实现,其中每个表有 256 个字节。

接着,在 MixColumns 中,  $\mathbb{F}_2^8$  中元素间的乘法,也就是说,  $c(x)$  和  $s(x)$  系数间的乘法,或者更准确地说,固定矩阵的元素与式(7.7.4)中列向量元素间的乘法,也可以通过“查表法”实现:  $z = x \cdot y$  (域乘法),这里  $x \in \{ '01', '02', '03' \}$  和  $y \in \mathbb{F}_2^8$ 。进一步注意到字节 '01' 只不过是域中的乘法单位元,也就是  $'01' \cdot y = y$ 。因此这个乘法表的实现(无论是在软件中还是在硬件中)只需要  $2 \times 256 = 512$  项,这并不比一个小学生必须背诵的表大多少。因此这个实现不仅很快,而且还能够减少定时分析攻击的危险。

在式(7.7.4)中的线性代数运算和它的逆也有一个快速的“固化”实现法,希望深入探索的读者请参阅[88]。

### 7.7.5 AES 对应用密码学的积极影响

AES 的引入反过来又为应用密码学带来了几个积极的变化。

首先,多重加密,例如三重 DES,随着 AES 的出现而成为不必要的了:加长和可变的密钥及 128、192 和 256 比特的数据分组长度为各种应用要求提供了大范围可选的安全强度。由于多重加密多次使用密钥,那么避免使用多重加密就意味着实用中必须使用的密钥数目的减少,因此可以简化安全协议和系统的设计。

其次,AES 的广泛使用将促进同样强度的新杂凑函数的出现。在某些情形下,分组密码加密算法与杂凑函数密切相关(见 10.3.1 节),分组密码加密算法经常被用来作为单向杂凑函数,这已经成为一种标准应用。UNIX<sup>①</sup> 操作系统[208]的登录认证协议就是一个著名的例子,我们将在 11.5.1 节中看到 UNIX 口令方案的实现中 DES 函数的典型“单向变换”用法。另一个利用分组密码加密算法来实现(加密的)单向杂凑函数的例子可以在 10.3.3 节中找到。实用中,杂凑函数也经常被用做分组密码算法生成密钥的伪随机数函数。由于 AES 可变、加长的密钥和数据分组长度,将需要类似长度的杂凑函数。然而,由于平方根攻击(生日攻击,见 3.6 节和 10.3.1 节),杂凑函数的长度应该是分组密码密钥或数据分组长度的两倍,因此将需要与 128、192 和 256 比特的 AES 长度相匹配的 256、384 和 512 比特输出长度的新杂凑函数。ISO/IEC 现在正在进行杂凑函数 SHA-256、SHA-384 和 SHA-512 的标准化工作[153]。

最后,像 DES 标准吸引了许多试图攻破该算法的密码分析家的注意,随之促进了分组密码分析的认识水平的发展一样,作为新的分组密码标准的 AES 也将引起分组密码分析中高水平研究兴趣的再次兴起,这必将使得该领域的认识水平得到进一步的提高。

## 7.8 运行的保密模式

分组密码是将消息作为数据分组来处理的(加密或解密)。通常大多数消息(也就是一个

<sup>①</sup> UNIX 是贝尔实验室的一个注册商标。

消息串)的长度大于分组密码的消息分组长度,长的消息被分成一系列连续排列的消息分组,密码一次处理一个分组。

在基本的分组密码算法之后紧接着设计了许多不同的运行模式。这些运行模式(除去其中平凡的情形)为密文分组提供了几个希望得到的性质,例如增加分组密码算法的不确定性(随机性),将明文消息添加到任意长度(使得密文长度不必与相应的明文长度相关),错误传播的控制,流密码的密钥流生成,等等。

然而,我们不要认为使用这些运行模式就可以将“教科书式密码”的分组密码转化为适于实用的分组密码,这一点将在以后的讨论中予以说明(特别地,在 7.8.2.1 节中我们将看到一个有效的攻击,它可用于攻击实际中广泛使用的几个协议)。

这里我们描述五个常用的运行模式,它们是电码本(ECB)模式、密码分组链接(CBC)模式、输出反馈(OFB)模式、密码反馈(CFB)模式和计数器(CTR)模式。我们的描述遵从 NIST 的最新推荐 [220]。

在描述中,我们将使用下面的记号:

- $\mathcal{E}()$ : 基本分组密码的加密算法;
- $\mathcal{D}()$ : 基本分组密码的解密算法;
- $n$ : 基本分组密码算法的消息分组的二进制长度(在所有我们考虑的分组密码中,明文和密文消息空间是一样的,所以  $n$  既是分组密码算法输入的分组长度,也是输出的分组长度);
- $P_1, P_2, \dots, P_m$ : 输入到运行模式中明文消息的  $m$  个连续分段;
  - 第  $m$  分段的长度可能小于其他分段的长度。在这种情况下,可对第  $m$  分段添加,使其与它分段长度相同;
  - 在某些运算模式中,消息分段的长度等于  $n$  (分组长度),而在其他的运算模式中,消息分段的长度是任意小于或等于  $n$  的正数;
- $C_1, C_2, \dots, C_m$ : 从运算模式输出的密文消息的  $m$  个连续分段;
- $\text{LSB}_u(B), \text{MSB}_v(B)$ : 分别是分组  $B$  中最低  $u$  位比特和最高  $v$  位比特;例如:

$$\text{LSB}_2(1010011) = 11, \text{MSB}_5(1010011) = 10100$$

- $A \parallel B$ : 数据分组  $A$  和  $B$  的链接;例如:

$$\text{LSB}_2(1010011) \parallel \text{MSB}_5(1010011) = 11 \parallel 10100 = 1110100$$

### 7.8.1 电码本模式(ECB)

加密(或解密)一系列连续排列的消息段的一个最直接方式就是将它们逐个分别加密(或解密)。在这种情况下,消息分段恰好是消息分组。由于类似于在电报密码本中指定码字,我们给了这个自然而简单的方法一个正式的名字:电码本模式(ECB)。ECB 模式定义如下:

**ECB 加密**  $C_i \leftarrow \mathcal{E}(P_i), i = 1, 2, \dots, m;$

**ECB 解密**  $P_i \leftarrow \mathcal{D}(C_i), i = 1, 2, \dots, m。$

ECB 模式是确定性的,也就是说,如果在相同的密钥下将  $P_1, P_2, \dots, P_m$  加密两次,那么输出的密文分组也是相同的。在应用中,数据通常有部分可猜测的信息,例如,薪水的数目就

有一个可猜测的范围。如果明文消息是可猜测的,那么由确定性加密方案得到的密文就会使攻击者通过使用试凑法猜测出明文。例如,如果知道由 ECB 模式加密产生的密文是一个薪水数字,那么攻击者只需少量的试验就可以恢复出这个数字。通常,我们不希望使用确定性密码,因此在大多数应用中不要使用 ECB 模式。

### 7.8.2 密码分组链接模式(CBC)

密码分组链接(CBC)运行模式是用于一般数据加密的一个普通的分组密码算法。使用 CBC 模式,输出是  $n$  比特密码分组的一个序列,这些密码分组链接在一起使得每个密码分组不仅依赖于所对应的原文分组,而且依赖于所有以前的数据分组。CBC 模式有下面的运算:

**CBC 加密** 输入:  $IV, P_1, \dots, P_m$ ; 输出:  $IV, C_1, \dots, C_m$ ;

$$C_0 \leftarrow IV;$$

$$C_i \leftarrow \mathcal{E}(P_i \oplus C_{i-1}), i = 1, 2, \dots, m;$$

**CBC 解密** 输入:  $IV, C_1, \dots, C_m$ ; 输出:  $P_1, \dots, P_m$ ;

$$C_0 \leftarrow IV;$$

$$P_i \leftarrow \mathcal{D}(C_i) \oplus C_{i-1}, i = 1, 2, \dots, m$$

第一个密文分组  $C_1$  的计算需要一个特殊的输入分组  $C_0$ , 习惯上称之为“初始向量”(IV)。IV 是一个随机的  $n$  比特分组, 每次会话加密时都要使用一个新的随机 IV, 由于 IV 可看做是密文分组, 因此无须保密, 但一定要是不可预知的。由加密过程我们知道, 由于 IV 的随机性, 第一个密文分组  $C_1$  被随机化, 同样, 依次后续的输出密文分组都将被前面紧接着的密文分组随机化, 因此, CBC 模式输出的是随机化的密文分组。发送给接收者的密文消息应该包括 IV。因此, 对于  $m$  个分组的明文, CBC 模式将输出  $m + 1$  个密文分组。

令  $Q_1, Q_2, \dots, Q_m$  是对密文分组  $C_0, C_1, C_2, \dots, C_m$  解密得到的数据分组输出, 则由

$$Q_i = \mathcal{D}(C_i) \oplus C_{i-1} = (P_i \oplus C_{i-1}) \oplus C_{i-1} = P_i$$

可知的确正确进行了解密。图 7.3 给出了 CBC 模式的图示。

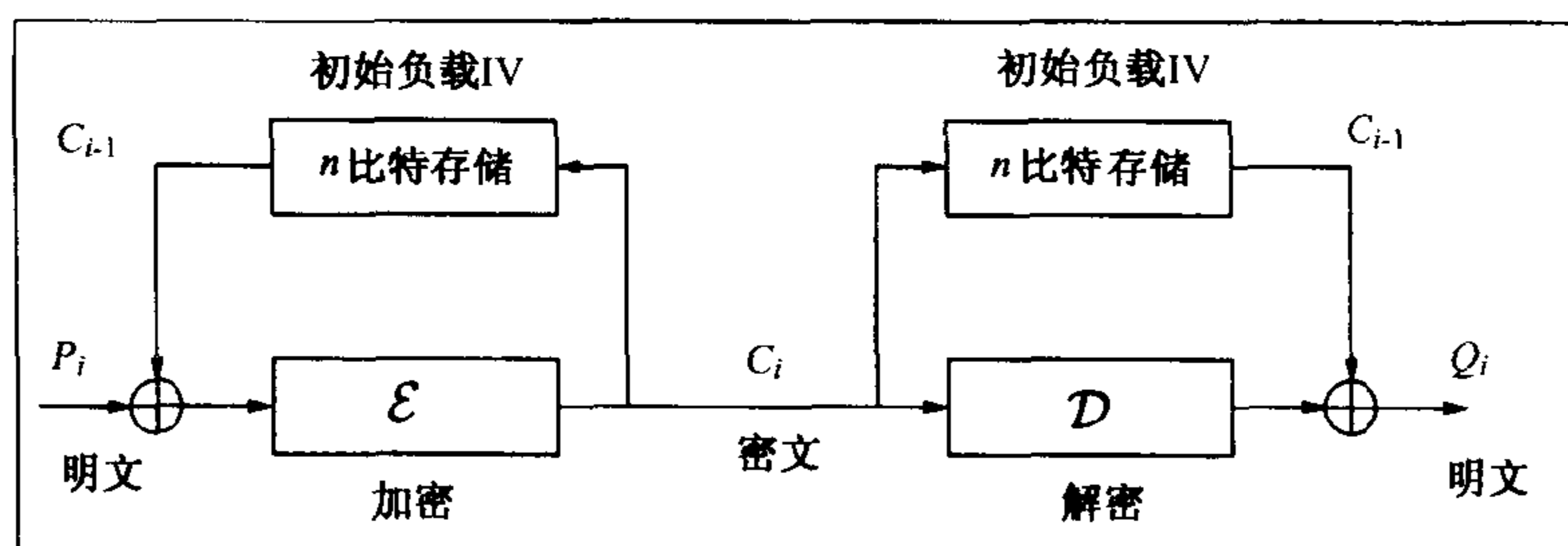


图 7.3 密码分组链接运行模式

#### 7.8.2.1 一般性误解

因为在 CBC 中, 数据分组是链接在一起的, 所以这种模式似乎可以提供保护, 防止非授权的数据篡改, 如删除、插入等(这种保护就是我们将在第 10 章中讨论的数据完整性)。一些分组密码算法因而指定使用 CBC 模式的算法方法作为工具来提供数据完整性。例如, RC5-CBC-PAD 模式[17]在进行 CBC 模式加密之前就指定用下面的 CBC 明文填充方案来处理明文消息分组:

1. 把明文消息串分成由字节(一个字节是 8 个比特)构成的序列,每 8 个消息字节构成一个(明文)消息分组(所以分组长度是 64)。
2. 最后 8 字节的明文消息分组一定是一个“填充分组”。前一部分是  $a$  个明文消息字节,  $0 \leq a \leq 7$ ,后一部分是  $8 - a$  个“填充字节”。每个“填充字节”都有固定的十六进制值  $8 - a$ 。例如,如果最后的消息分组有 7 个明文消息字节,这些消息字节后面跟着一个“填充字节”‘01’,那么填充分组就是

$$\text{消息字节}_1 \parallel \text{消息字节}_2 \parallel \cdots \parallel \text{消息字节}_7 \parallel '01'$$

然而,如果最后的消息分组只有一个明文消息字节,那么填充分组就是

$$\text{消息字节} \parallel '07' \parallel '07' \parallel '07' \parallel '07' \parallel '07' \parallel '07' \parallel '07'$$

如果消息字节的数目可被 8 整除,那么消息字节后面跟着如下所示的全部由“填充字节”构成的填充分组:

$$'08' \parallel '08' \parallel '08' \parallel '08' \parallel '08' \parallel '08' \parallel '08' \parallel '08'$$

其他的 CBC 加密方案使用类似的填充方案。例如,在 IPSec[164](将在第 12 章中介绍)使用的“IP 封装安全载荷”(ESP)中, $X$ “填充字节”(  $1 \leq X \leq 255$ )是

$$'01' \parallel '02' \parallel \cdots \parallel 'xy'$$

这里  $'0' \leq 'x' \leq 'F'$  和  $'0' \leq 'y' \leq 'F'$ ,符号  $'xy'$  是整数  $X$  的十六进制表示。解密时,重新得到的明文消息中显示的“填充字节”将被删掉(当然是在检查完“数据完整”一致性之后)。

国际标准化组织(ISO)[146,147]两个早期文献草案中的几个认证协议也曾建议使用 CBC 加密模式来提供“数据完整性保护”(这些协议使用 CBC 的一般指南记录在文件[148,144]中)。

然而,事实上在任何意义上认为 CBC 能够提供数据完整性保护都是完全错误的。

对于 CBC“填充字节”方案,如果利用该方案的目的就是要提供数据完整性保护,那么 Vaudenay 演示了一种攻击[296],该攻击表明了这种保护是不存在的。在 Vaudenay 的攻击中, Malice(攻击者)发送给一个参与者(密钥持有者,称为解密预言机<sup>①</sup>,提供预言服务)两个适应性操纵的密文分组

$$r, C_i$$

其中  $r$  是任意的数据分组,  $C_i = \mathcal{E}(P \oplus C_{i-1})$  是密文分组, Malice 想要知道关于它的相应的明文消息  $P$ (例如,  $P$  是口令)的信息。从“CBC 解密”我们知道相应的解密将是

$$P \oplus C_{i-1} \oplus r$$

“数据完整性”检验法将指示解密预言机如何操作。由解密预言机的操作, Malice 可以有很好的机会得出有关明文消息  $P$  的某些信息。例如,如果“数据完整性保护机制”指令解密预言机一旦看到一个“有效填充”就回答 YES,那么最可能的“有效填充”就是最后的“填充字节”是‘01’时的情形。因为概率空间是一个有 8 个字节的比特,所以这种情况发生的概率接近于  $2^{-8}$ 。正是在这种情形下,因为  $r$  的随机性,除“正确填充”以外的其他情形使解密预言机回答 YES 的概率要低得多(由于两个或更多字节的概率空间要大得多),因此可以忽略。那么 Malice 就发现了

① 术语“预言”经常出现于密码学书面语中,通常用来命名那些声称能够解困难问题的任何未知的算法或方法。预言服务是指用户使用攻击者得不到的密钥为攻击者提供(经常是不经意地)密码运算。



$$\text{LSB}_8(P) = \text{LSB}_8(r) \oplus '01'$$

也就是说, Malice 已经成功地恢复了  $P$  的最后一比特, 这是有关  $P$  的重要信息!

如果解密过程检测到发生了填充错误(其概率接近于  $1 - 2^{-8}$ , 理由见上面), 预言机可以直接回答 NO, 也可以根本不做回答(程序终止, 仿佛预言机爆炸了, 因此 Vaudenay 将这种预言机命名为**炸弹式预言机**)。但是, “没有回答”实际上就是一种回答, 在这种情况下, 回答就是 NO。在回答是 NO(直接或含蓄)的情况下, Malice 不能得到最后一字节的任何信息, 但是他可以改变  $r$ , 重新尝试, 这是针对提供预言服务的主体的一种**主动攻击**。我们将在 8.6 节中正式定义主动攻击。关于起预言服务提供者作用的参与者的更多方案, 我们将在本书其余部分的许多地方看到。

Vaudenay 将他的攻击技术用到了许多实际中得到广泛应用的几个密码协议上, 例如 IPsec、SSH 和 SSL(这些协议将在第 12 章中介绍)。在这些实用中, 即使答案并不以明确的方式给出(例如, 答案是加密的), Malice 也会很容易得到 YES/NO 回答。

在这个攻击的基本形式中, 如果解密预言机“不爆炸”, 它只以相当小的概率  $\approx 2^{-8}$  回答最后一字节。不过在许多应用中相当标准的设置下, 有许多方法来保持预言机不爆炸, 所以它能够进一步回答问题来使得 Malice 进一步获得明文字节的信息。假定在关于最后的明文字节给出了一个回答 YES 以后, 预言机仍然可用, 那么 Malice 能够将  $r$  改为  $r'$ , 使得

$$\text{LSB}_8(r') \leftarrow \text{LSB}_8(r) \oplus '01' \oplus '02'$$

然后将  $r', C$  发送给预言机, Malice 就能够以同样的概率  $2^{-8}$  来得到关于倒数第二个明文的信息。如果预言机能够保持不爆炸, 攻击者就可以继续下去, 从而 Malice 就可以从  $8 \times 2^8 = 2048$  次预言回应中得到整个明文分组的信息。

在 12.5.4 节中我们将看到, Vaudenay 的攻击被用于使用 SSL/TLS 协议的电子邮件应用中的 CBC 明文填充实现。在那个攻击中, 解密预言机就是一个永不爆炸的电子邮件服务器, 因此可以让 Malice 得到整个明文消息分组的信息, 这些明文消息就是用户读取电子邮件的口令。该攻击利用了可通过**定时分析**得到的**边信道**信息, 因此这种攻击称为**边信道攻击**。

利用 CBC 来提供数据完整性保护的 ISO 协议也有致命的缺陷 [186, 187]。在 17.2.1.2 节中我们将通过分析使用标准的 CBC 实现加密的认证协议来说明该缺陷。设计这个协议时希望通过使用 CBC 可以提供密码上的数据完整性保护, 然而, 这个协议的缺陷恰恰就是没有这项服务。

使输出密文随机化看来是 CBC 模式提供的仅有安全服务, 要保证 CBC 的密文输出的数据完整性, 必须要借助于我们将要在第 10 章讨论的其他密码技术。

### 7.8.2.2 一个警告

Knudsen 观察到 CBC 中的一个保密性缺陷 [167], 描述如下。当两个密文  $C_i, C'_j$  分组相等时, 那么从 CBC 加密算法我们可以得到

$$C_{i-1} \oplus C'_{j-1} = P_i \oplus P'_j$$

由于明文通常包含冗余, 这个方程就有助于由密文恢复明文, 这可为窃听者所利用。所以为了使得利用这个方程的攻击不可行, 我们在每次重新加密时总是要使用随机的 IV, 这样两个密文相等的概率就小得可以忽略了(随机的 IV 提供了一个非常大的概率空间)。



### 7.8.3 密码反馈模式(CFB)

密码反馈(CFB)运行模式的特点在于反馈相继的密码分段,这些分段从模式的输出返回作为基础分组密码算法的输入。消息(明文或密文)分组长为  $s$ , 其中  $1 \leq s \leq n$ 。CFB 模式要求 IV 作为初始的  $n$  比特随机输入分组,因为在系统中 IV 是在密文的位置中,所以它不必保密。CFB 模式有下面的运算:

**CFB 加密** 输入:  $IV, P_1, \dots, P_m$ ; 输出:  $IV, C_1, \dots, C_m$ ;

$$I_1 \leftarrow IV;$$

$$I_i \leftarrow \text{LSB}_{n-s}(I_{i-1}) \parallel C_{i-1} \quad i = 2, \dots, m;$$

$$O_i \leftarrow \mathcal{E}(I_i) \quad i = 1, 2, \dots, m;$$

$$C_i \leftarrow P_i \oplus \text{MSB}_s(O_i) \quad i = 1, 2, \dots, m。$$

**CFB 解密** 输入:  $IV, C_1, \dots, C_m$ ; 输出:  $IV, P_1, \dots, P_m$ ;

$$I_1 \leftarrow IV;$$

$$I_i \leftarrow \text{LSB}_{n-s}(I_{i-1}) \parallel C_{i-1} \quad i = 2, \dots, m;$$

$$O_i \leftarrow \mathcal{E}(I_i) \quad i = 1, 2, \dots, m;$$

$$P_i \leftarrow C_i \oplus \text{MSB}_s(O_i) \quad i = 1, 2, \dots, m。$$

观察到在 CFB 模式中,基本分组密码的加密函数用在加密和解密的两端,因此,基本密码函数  $E$  可以是任意(加密的)单向变换,例如单向杂凑函数。CFB 模式可以考虑作为流密码的密钥流生成器,加密变换是作用在密钥流和消息分段之间的弗纳姆密码。类似于 CBC 模式,密文分段是前面所有的明文分段的函数值和 IV,图 7.4 给出了 CFB 模式的图示。

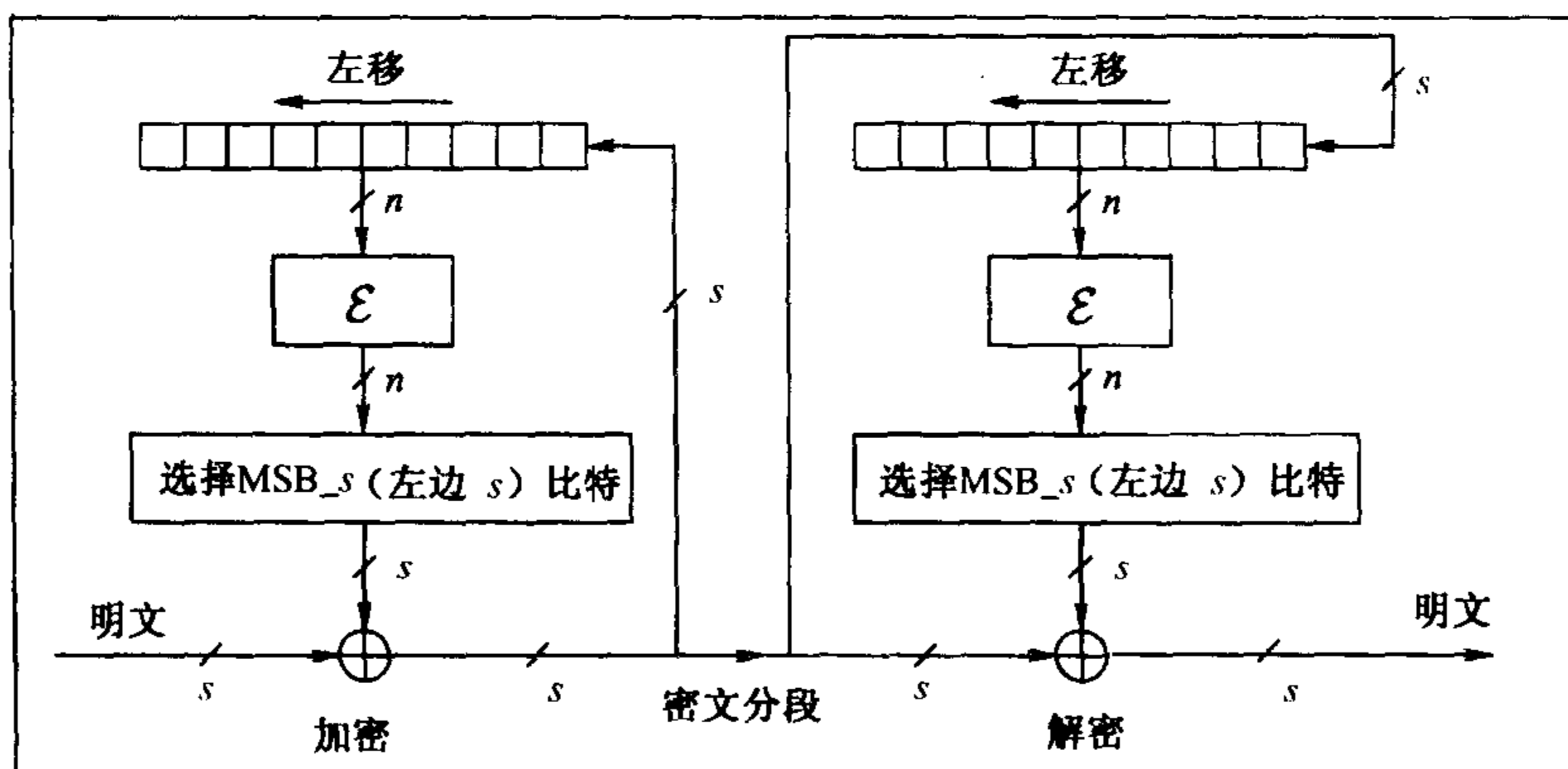


图 7.4 密码反馈运行模式

### 7.8.4 输出反馈模式(OFB)

输出反馈(OFB)运行模式的特点是将基本分组密码的连续输出分组回送回去。这些反馈分组构成了一个比特串,用做弗纳姆密码的密钥流的比特串,就是密钥流与明文分组相异或。OFB 模式要求 IV 作为初始的随机  $n$  比特输入分组。因为在系统中,IV 是在密文的位置中,所以它不需要保密。OFB 模式运算如下:

**OFB 加密** 输入:  $IV, P_1, \dots, P_m$ ; 输出:  $IV, C_1, \dots, C_m$ ;

$$I_1 \leftarrow IV;$$

$$I_i \leftarrow O_{i-1} \quad i = 2, \dots, m;$$

$$O_i \leftarrow \mathcal{E}(I_i) \quad i = 1, 2, \dots, m;$$

$$C_i \leftarrow P_i \oplus O_i \quad i = 1, 2, \dots, m。$$

**OFB 解密** 输入:  $IV, C_1, \dots, C_m$ ; 输出:  $IV, P_1, \dots, P_m$ ;

$$I_1 \leftarrow IV;$$

$$I_i \leftarrow O_{i-1} \quad i = 2, \dots, m;$$

$$O_i \leftarrow \mathcal{E}(I_i) \quad i = 1, 2, \dots, m;$$

$$P_i \leftarrow C_i \oplus O_i \quad i = 1, 2, \dots, m。$$

在 OFB 模式中,加密和解密是相同的:将输入消息分组与由反馈电路生成的密钥流相异或。反馈电路实际上构成了一个有限状态机,其状态完全由基础分组密码算法的加密密钥和 IV 决定。所以,如果密码分组发生了传输错误,那么只有相应位置上的明文分组会发生错乱,因此,OFB 模式适宜不可能重发的消息加密,如无线电信号。类似于 CFB 模式,基础分组密码算法可用加密的单向杂凑函数代替。图 7.5 给出了 OFB 模式的图示。

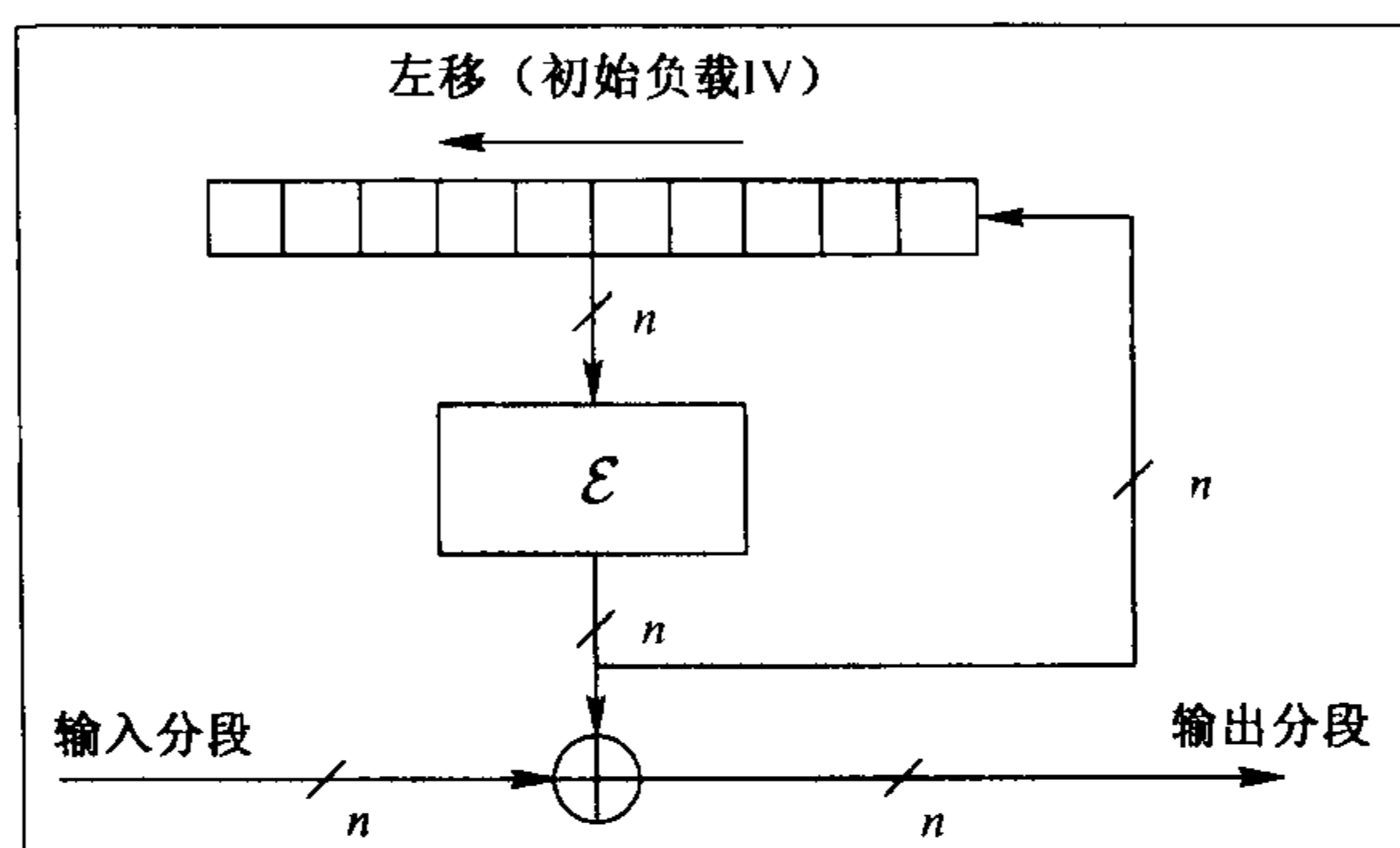


图 7.5 输出反馈运行模式(加密和解密)

### 7.8.5 计数器模式(CTR)

计数器(CTR)模式的特征是将计数器从初始值开始计数所得到的值馈送给基础分组密码算法。随着计数的增加,基础分组密码算法输出连续的分组来构成一个比特串,该比特串被用作弗纳姆密码的密钥流,也就是密钥流与明文分组相异或。CTR 模式运算如下(这里  $Ctr_1$  是计数器初始的非保密值):

**CTR 加密** 输入:  $Ctr_1, P_1, \dots, P_m$ ; 输出:  $Ctr_1, C_1, \dots, C_m$ ;

$$C_i \leftarrow P_i \oplus \mathcal{E}(Ctr_i) \quad i = 1, 2, \dots, m$$

**CTR 解密** 输入:  $Ctr_1, C_1, \dots, C_m$ ; 输出:  $Ctr_1, P_1, \dots, P_m$ ;

$$P_i \leftarrow C_i \oplus \mathcal{E}(Ctr_i) \quad i = 1, 2, \dots, m$$

因为没有反馈,CTR 模式的加密和解密能够同时进行,这是 CTR 模式比 CFB 模式和 OFB 模式优越的地方。由于这种模式很简单,我们省略了 CTR 模式的图示。

## 7.9 对称密码体制的密钥信道建立

在通信双方能够使用对称密码体制开始保密通信以前,必须首先生成它们之间共享的正

确密钥。这里,“正确”不仅是指所生成的密钥是逐比特正确的,就是说没有错误,还指双方都必须确信与之分享密钥的一定是意定通信者。

一个通信信道,通过它正确建立密钥,该通信信道就称为密钥信道(见图 7.1)。密钥信道与消息信道是分开的,二者之间的不同在于密钥信道是要保护的,而通信信道是不受保护的。在对称密码体制中,因为加密密钥与解密密钥相同,所以密钥信道既要保证密钥的保密性,也要保证它的可靠性。

可以通过三种方式建立对称密码体制的密钥信道:传统技术、公钥技术和量子密钥分配(QKD)技术。

**传统技术** 在建立系统的时候,物理上安全的方式,例如,快信传递业务,可以用来使两个用户专有地分享初始密钥。通常,这两个用户中有一个是可信赖第三方(TTP),他将提供可认证服务(信赖的含义见 2.4 节)。一旦一个初始密钥由一个终端用户和一个 TTP 分享,TTP 是一个长期的密钥信道,那么任意两个终端用户都可以运行一个认证协议来确保在他们之间建立一条安全的密钥信道。使用 TTP 减少了终端用户密钥管理的负担:如果在任意两对终端用户之间存在长期的密钥信道,那么终端用户就不必管理许多密钥。在第 2 章中,我们已经看到几个认证和密钥建立协议的例子,使用终端用户和认证服务器之间的长期密钥信道为用户双方建立会话密钥。在第 11 章、第 12 章和第 17 章研究认证协议、系统及其安全性分析的形式化方法时,我们将看到更多这样的协议。

传统密钥信道建立技术的一个严重缺点就是它必须依赖于在线认证服务,这一缺点限制了该技术在任何开放系统中的应用。事实上,迄今为止这种技术只在一个企业环境中得到了很好的应用,我们将在 12.4 节中给出该应用的详细讨论。

**公钥技术** 公钥密码学的一个重要优点就是易于建立两个相距遥远的终端用户间的密钥信道,而不需要他们彼此见面或者使用在线认证服务,这正好克服了传统技术的缺点。因此,基于公钥的技术能够容易地增加在大的开放性系统中的应用。有许多用于密钥信道建立的公钥技术。我们将在下一章中介绍公钥密码学,并在第 13 章中研究基于公钥技术的认证框架。

然而,有了公钥密码学,仍有必要建立从用户到系统的安全密钥信道。这里,“安全”是指认证:可以确定给定的公钥是否确实为声称的用户所拥有。然而,使用公钥技术的密钥信道建立并不需要处理任何秘密,关于公钥的密钥信道的建立确实是一个纯认证问题。在图 7.1 中,我们已经演示了公钥信道可基于目录服务。我们将在第 12 章的 12.3 节讨论用于建立公钥认证信道的一些实用认证技术,在第 13 章中讨论用于建立公钥认证框架的一般技术。

**量子密钥分配技术** 在 4.4.5.1 节我们已经看到了一项用于获得量子密钥分配(QKD, 协议 4.1)的技术。QKD 协议允许两个用户尽管实际上从未会面却可以协商密钥。类似于公钥技术的例子,仍然有必要建立一个由用户到系统的认证信道,该认证信道可以基于某些单向函数,使得一个终端用户拥有单向函数的一个秘密原像,可以允许他的通信伙伴验证,但是前者并不将秘密泄漏给后者。利用认证信道,QKD 协议的参与者可以确保协议是在其与意定通信伙伴之间进行的,商业 QKD 系统预计将在 2004 年左右投入使用[270]。

我们必须强调指出用于密钥信道建立的 QKD 技术前景的重要性。绝大多数实用的基于复杂性理论的公钥技术(基于找到周期函数的周期的困难性)将受到实用量子计算技术的挑战。然而,QKD 技术是量子技术免疫的(看起来存在非周期的单向函数,这些单向函数是量子技术免疫的并且可用于认证的目的)。因此,即使量子计算技术成为实际有效的,QKD 技术也

将用于密钥信道建立,使得分享密钥的双方不必实际会面或依赖于由可信赖第三方提供的在线认证。

最后,我们将注意到基于公钥的技术和 QKD 技术说明了通过完全公开的讨论能够建立保密的通信信道,这是众所周知的原则(见[190,191])。

## 7.10 本章小结

本章中我们讨论了对称加密算法的原理,介绍了几个对称加密方案。

首先我们介绍了古典密码,根据香农信息论,我们考虑了它们的有条件的安全性,并指出了古典密码的工作原理:代换,仍是现代对称加密算法

介绍了两个现代分组加密算法:DES和AES。介绍了DES的历史地位和仍在使用的它的Feistel密码设计结构的有效性。描述了作为最新建立的加密标准AES,并详细解释了它的工作原理。我们也考虑了AES快速和安全的实现方法,并且讨论了AES对应用密码学可能有的积极影响。

然后我们介绍了使用分组密码所用的各种标准的运行模式,讨论了普通的运行模式CBC,并揭示出了一个共同的误解,就是CBC能提供数据完整性业务,我们已经论证了这是错误的。在第17章我们研究使用CBC加密的认证协议时,将给出这一误解的更为清楚的证据。

最后,我们列出了三种技术,为希望传送保密信息的通信方之间建立安全密钥信道。其中,QKD技术尽管仍处于雏形,但由于它不受量子计算技术的影响,未来将是极其重要的。

## 习题

- 7.1 加密算法为什么不应该包含秘密设计部分?
- 7.2 明文在整个消息空间中的一个很小的区域内,英语中某些字符不相等的频率就是这样的一个例子。再给出两个也能说明英语明文消息有小的区域分布的例子。
- 7.3 令  $S_p$  和  $S_c$  分别表示明文消息源和相应的密文消息源,利用 3.7 节中给出的熵公式,解释由简单代换或换位密码得到的密文消息输出没有改变相应明文消息的分布,或者说,密文仍然在整个消息空间的一个小区域内。  
提示:  $H(S_p) = H(S_c)$ 。
- 7.4 弗纳姆密码是一种代换密码吗?它是单表代换还是多表代换?
- 7.5 弗纳姆密码和一次一密的不同之处是什么?
- 7.6 为什么说一次一密加密抗窃听是无条件安全的?
- 7.7 由于协议 7.1 中的移位密码使用了一次一密,密钥长度与消息长度相同,所以它是一个完善的安全加密方案。如果移位密码的计算是加法无模约简,它还是一个完善的安全加密方案吗?
- 7.8 虽然简单代换密码和换位密码对频度分析攻击是十分脆弱的,为什么它们仍被广泛使用在现代加密方案和密码协议中?

- 7.9 现代密码通常是由几个古典密码技术结合起来构造的。在 DES 和 AES 中找出采用了下述三种密码技术的部分：(i) 代换密码，(ii) 换位密码，(iii) 弗纳姆密码。
- 7.10 (i) 为什么 AES 被认为是非常有效的？(ii) 在 AES 的实现中，有限域  $\mathbb{F}_2^8$  中的乘法是如何实现的？
- 7.11 在分组密码的密码分组链接(CBC)运行模式下，如果收到的密文的解密“具有正确的填充”，你认为传输的明文有有效的数据完整性吗？

## 第8章 加密——非对称技术

### 8.1 引言

早期密码(如恺撒密码)依赖于对整个加密过程的保密。现代密码,像 DES 和 AES 遵循 Kerchoffs 原则(见 7.1 节):公开算法的细节来公开验证这些密码的安全性。密码设计者这样做的目的是希望证明他们所设计的密码的安全性仅依赖于加密密钥的选择。

还可以进一步应用 Kerchoffs 原则在加密算法中减少需要保密的成分。考虑香农关于加密的语意性质:一个混合变换,它把明文空间 $\mathcal{M}$ 中有意义的消息均匀地分布到整个消息空间 $\mathcal{C}$ 中去[见第 7 章的式(7.1.1)]。我们知道不必运用任何秘密便可以得到这样的随机分布。1975 年,Diffie 和 Hellman 首先实现了这一点[98](这篇文章的出版时间是 1976 年,但是在 1975 年 9 月,这篇文章作为预印稿公布,见[97])。他们把这个发现称为**公钥密码学**。在那个时代,它是对密码学的一个全新的理解。

在公钥密码体制中,加密不用秘密钥,秘密钥仅在解密阶段使用。在文[98]中,Diffie 和 Hellman 描述了几个可能用来实现公钥密码的数学变换,它们称之为**单向陷门函数**。非正式地讲,单向陷门函数有以下性质:

**性质 8.1 单向陷门函数** 单向陷门函数  $f_t(x): \mathcal{D} \rightarrow \mathcal{R}$ , 是一个单向函数,即对任意的  $x \in \mathcal{D}$ , 容易计算,而对几乎所有的  $x \in \mathcal{D}$ , 求逆困难。但是,如果知道陷门信息  $t$ ,则对所有的  $y \in \mathcal{R}$ , 容易计算满足  $y = f_t(x)$  的  $x \in \mathcal{D}$ 。

单向陷门函数的概念使得公钥密码系统成为可能。相对于私钥(或者说对称)密码系统,由于单向陷门函数的非对称性,基于单向陷门函数的公钥密码系统被称为**非对称密码系统**。虽然在 Diffie 和 Hellman 关于公钥密码体制的第一篇论文中提到的几种单向陷门函数([98])不具有很好的非对称性,因而不能用于公钥体制,但是 Diffie 和 Hellman 不久便提出了一个很好的函数:模指数运算,并用它来演示这个著名的密码协议:**Diffie-Hellman 密钥交换协议**[99](见 8.3 节)。目前,这个公钥算法仍然在广泛地使用并且也在继续不断地向前发展。

在 1974 年,Merkle 发现了一种通过非对称计算来实现密钥协商的机制,称为 **Merkle 难题**[201]。Merkle 难题的非对称计算意味着密钥协商协议的合法参与者与搭线窃听者的计算复杂度有很大的差别:前者是可行的,后者是不可行的。Merkle 难题第一次实现了单向陷门函数。虽然 Merkle 难题可能不适应于现代密码的应用(由于非对称性介于  $n$  和  $n^2$  之间),但它所揭示的思想是发现公钥密码学的里程碑。

1973 年,英国密码学家 Cocks 构造了第一例公钥密码系统(参阅[279])。Cocks 的加密算法(称为“无密钥加密”)基于大整数分解并且本质上和 RSA 相同(见 8.5 节)。遗憾的是,Cocks 加密算法被保密起来。直到 1997 年 12 月,英国政府的通信服务电子安全工作组(CESG)才公布 Cocks 算法。



虽然在封闭部门知道公钥密码体制这一概念之后,公开研究团体才发现了公钥密码体制,我们必须指出,正是公开研究的团体发现了公钥密码的最重要的两个应用:(i) 数字签名(见 10.4 节),(ii) 公开信道的密钥建立(见 8.3 节)。这两个应用促进了当今因特网上安全电子商务的日益增长。

### 8.1.1 本章概述

我们首先从本章的技术方面,引入“教科书式密码”安全性的概念,提前给出一个警告:从现实中的标准应用来看,本书介绍的所有公钥加密算法实际上都是不安全的(8.2 节)。我们接着介绍几个著名的公钥密码原型,这些原型是:Diffie-Hellman 密钥交换协议(8.3 节),教科书式 RSA(8.5 节)、Rabin(8.10 节)和 ElGamal(8.12 节)密码体制。在介绍这些基本的公钥密码原型的同时,也在相应隐含的困难问题假设的基础上介绍形式理论和复杂性理论。这些假设是:Diffie-Hellman 问题和离散对数问题(8.4 节)、RSA 问题(8.7 节)和整数分解问题(8.8 节)。我们也将在本章开始引入形式化概念来描述对公钥体制的不同攻击类型(8.6 节)。在 8.9 节(RSA)、8.11 节(Rabin)、8.13 节(ElGamal)中将举例说明教科书式密码算法的不安全性。我们将考虑对公钥加密的更安全定义的需要(8.14 节)。在介绍对称和非对称密码系统之后,我们将介绍它们的组合:混合加密体制(8.15 节)。

## 8.2 “教科书式加密算法”的不安全性

我们应该注意这一章所介绍的加密算法应该归为教科书式密码这一类。这样归类是因为这些算法可以在关于密码学的大多数教科书中找到。但是,这些基本的加密算法实际上不适合具体的应用。从公钥密码系统角度来看,一般的教科书式加密算法具有性质 8.2 所陈述的保密特性。

### 性质 8.2 教科书式加密算法的不安全性

在这一章,一个加密系统的安全性(保密性)从以下两个方面来考虑:

- i) **完全或无保密** 已知加密算法及其输出的一条密文,攻击者的任务是恢复出这条长度通常由密码系统的安全参数决定的整条明文;或者是在一个已知加密算法的基础上,给定一组明-密文对,攻击者的任务是恢复整个密钥。攻击者或者成功地完全得到想要的秘密,或者什么也得不到。我们应该特别注意“无”的含义:它意味着无论在攻击之前或者之后,攻击者没有得到关于秘密的任何信息。
- ii) **被动攻击** 攻击者不能运用自己掌握的数据操纵或修改密文,也不能要求拥有密钥的用户提供加密或解密服务。

这种安全性(保密性)的定义非常弱。实际上,没有多大用处,所以称之为“不安全性定义”还算比较合适。

让我们来解释一下为什么性质 8.2(i) 是一条不安全性质。在应用中,明文数据很可能含有一些非秘密的“部分信息”,攻击者可以知道这些信息。例如,某些数据永远在一个比较小的范围之内:通常的工资数据会小于 1 000 000,即使是很高的工资,从密码角度来讲也是很小的一个数字。再如,通常的口令是一条由 8 个字母组成的比特串。通常知道了这个“部分信息”攻击者就能成功地得到所有的明文消息,而不是“一无所获”。



现在进一步解释为什么性质 8.2(ii)也是一条不安全性质。我们永远都不要期望攻击者是善意和被动的。典型的攻击者会运用他所能使用的任何手段。这里尤其包括攻击者参与目标用户的信息交互,发送密文给目标用户解密,得到相应的明文。这种交互的方式称为用户(公钥的拥有者)为攻击者提供**预言解密服务**。我们将在这一章以及以后的几个章节中看到,很难避免不提供这样的预言服务。

教科书式密码算法通常拥有良好的代数性质,这一点经常使得到服务的攻击者能够攻破教科书式密码算法。在这一章,我们将看到几个这样的例子,并且在以后的章节中也会看到这些攻击技巧的推广运用。

在这一章,我们将经常提醒用户不要提供预言服务,我们也应该注意到公钥算法的普通用户不知道如何才能不为攻击者提供预言服务。并且,避免被用做预言机也是一个很困难的问题(我们将在 12.5.4 节看到这一点)。正确的对策就是设计一个适用于应用的密码系统,普通的用户也可以安全地使用。

通过陈述性质 8.2,我们具体阐明了在本章范围内,不考虑关于公钥加密算法更安全的定义;而对于这里要介绍的教科书式加密算法,我们也不期望它在很强的意义下也是安全的。相反,我们将指出,教科书式加密算法在两条不安全性特征方面的一些保密缺陷,即部分信息泄漏和/或主动攻击的后果,在这里我们不准备修补这些缺陷。

第 14 章将介绍有关更严格的安全定义来抵抗更强(或更实际)的攻击。在第 15 章将介绍适于应用的与教科书式加密算法所对应的加密算法。

### 8.3 Diffie-Hellman 密钥交换协议

对于对称密码系统,在进行保密通信之前,必须向通信双方分别递送一个密钥。在公钥密码体制出现之前,通信双方建立共享密钥一直是一个比较困难的问题,这是因为这个过程需要一条安全的信道,通常这样的信道意味着要由专门的信使以物理方式传送。与对称密码体制相比,公钥密码体制的一个显著优点就是远程通信各方可以无需安全信道就能实现相互交换密钥。最早实现这一点的一个实用方案由 Diffie 和 Hellman 提出,称为 Diffie-Hellman 指数密钥交换协议[99]。

首先,假设用户 Alice 和 Bob 约定有限域  $\mathbb{F}_q$  和元素  $g \in \mathbb{F}_p$ ,  $g$  生成一个高阶群。为简化起见,我们考虑有限域  $\mathbb{F}_p$ ,  $p$  为素数时的情况,即  $\mathbb{F}_p$  是一个素域。双方可以用算法 4.5 来对  $p$  做素性检测,其中他们所构造的素数  $p$  满足  $p-1$  的完全分解是知道的;接着用算法 5.1 找出生成元  $g$ (即  $\mathbb{F}_p^*$  的生成元)。由定理 5.11,  $[1, p)$  中的每个数均可表示为  $g^x \pmod{p}$  形式,其中  $x$  为某个整数。在协议 8.1 中具体给出的基本 **Diffie-Hellman 密钥交换协议**中,  $p$  和  $g$  是参与者共同的输入。

由协议 8.1 不难看出,对于 Alice 有

$$k = g^{ba} \pmod{p}$$

对于 Bob 有

$$k = g^{ab} \pmod{p}$$

我们注意到由于  $ab \equiv ba \pmod{p-1}$ , 这样双方计算得到的值是相同的。这就是 Diffie-Hellman 密钥交换协议在通信双方之间实现了一个共享密钥的原因。

系统范围内的所有用户可以共用公开参数  $p$  和  $g$ 。

### 协议 8.1 Diffie-Hellman 密钥交换协议

共同输入  $(p, g)$ :  $p$  为大素数,  $g$  为  $\mathbb{F}_p^*$  的生成元。

输出 Alice 和 Bob 共享  $\mathbb{F}_p^*$  中的一个元素。

1. Alice 选择  $a \in_U [1, p-1]$ ; 计算  $g_a \leftarrow g^a \pmod{p}$ ; 发送  $g_a$  给 Bob;
2. Bob 选择  $b \in_U [1, p-1]$ ; 计算  $g_b \leftarrow g^b \pmod{p}$ ; 发送  $g_b$  给 Alice;
3. Alice 计算  $k \leftarrow g_b^a \pmod{p}$ ;
4. Bob 计算  $k \leftarrow g_a^b \pmod{p}$ 。

例 8.1 令  $p = 43$ , 应用算法 5.1 我们知道 3 是  $\text{mod } 43$  的一个本原根。令 Alice 和 Bob 知道共同的公开参数  $(p, g) = (43, 3)$ 。

为了协商一个密钥, Alice 随机选取她的秘密指数 8, 把  $3^8 \equiv 25 \pmod{43}$  发送给 Bob。Bob 随机选取他的秘密指数 37, 把  $3^{37} \equiv 20 \pmod{43}$  发送给 Alice。他们所协商的密钥就是

$$9 \equiv 20^8 \equiv 25^{37} \pmod{43} \quad \square$$

在执行和应用 Diffie-Hellman 密钥交换协议的时候, 我们应该注意下面一些细节:

- 共同的输入  $p$  应该为一个素数(或素数的幂), 满足  $p-1$  有足够大的素因子  $p'$ ; 这里“足够大”意味着  $p' > 2^{160}$ 。需要  $p$  具有这样的性质的原因将在 8.4 节中讨论。
- 共同的输入  $g$  不必是  $\mathbb{F}_p^*$  的生成元, 但  $g$  应该是  $\mathbb{F}_p^*$  中高阶子群的一个生成元, 例如阶为  $p'$  的子群。这种情形下, Alice 和 Bob 应验证  $g \neq 1$  和  $g^{p'} \equiv 1 \pmod{p}$ 。这样的话,  $p'$  也应该是这个协议共同输入的一部分。
- Alice(Bob)应该验证  $g_b \neq 1$  ( $g_a \neq 1$ )。对于他们各自取自  $(1, p')$  中的指数, 这个验证将确保共享的密钥  $g^{ab}$  是  $\mathbb{F}_p$  的  $p'$  阶子群中的一个元素, 也就是说, 是一个高阶子群中的一个元素。
- 在这个协议结束之后, Alice(Bob)应该立即删除她的指数  $a$  (他的指数  $b$ )。这样做, 在通信结束后, 如果他们正确地处理了交换密钥  $g^{ab}$ , 对于这个交换密钥, 他们将会拥有前向保密的性质。我们将在 8.15 节和 11.6.1 节进一步讨论“前向保密”性。

### 8.3.1 中间人攻击

应该注意, Diffie-Hellman 密钥交换协议不支持对协商密钥的认证功能。处于 Alice 和 Bob 通信中间的主动攻击者能够操纵这个协议的消息以达到成功的攻击, 称为中间人攻击。攻击 8.1 举例说明了这种攻击。

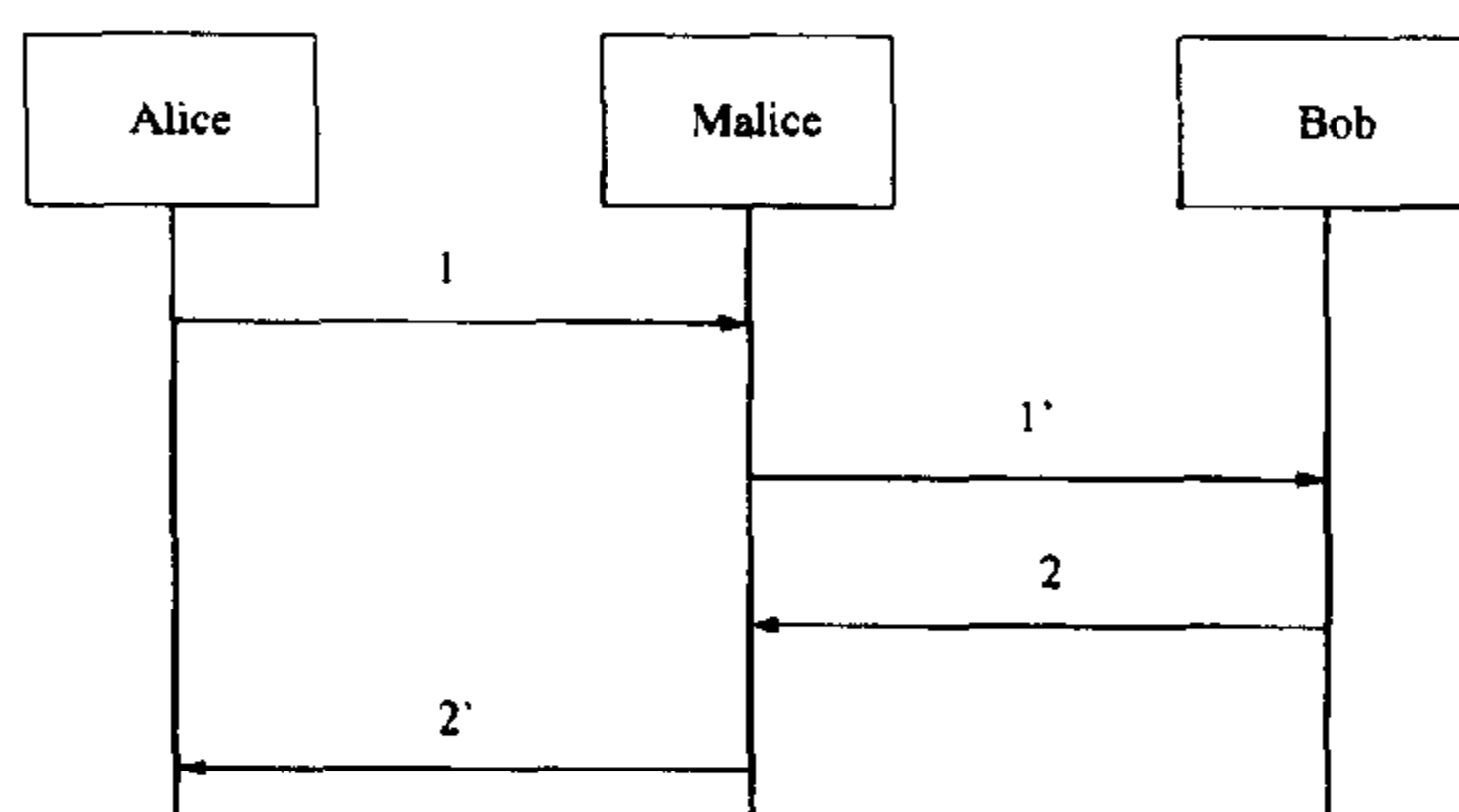
在对协议运行的攻击当中, Malice(坏家伙)截获 Alice 发送给 Bob 的第一条消息  $g_a$ , 并伪装成 Alice 向 Bob 发送消息

Malice(“Alice”)发送给 Bob:  $g_m \stackrel{\text{def}}{=} g^m \pmod{p}$

(读者可以回忆在 2.6.2 节中,我们习惯上把 Malice 的伪造行为表示为另外的主体)Bob 将按照协议的规则回复  $g_b$  给 Malice(“Alice”)。这意味着这个发送的数值再一次被 Malice 截获。现在 Malice 和 Bob 协商了一个密钥  $g^{bm}(\bmod p)$ ,而 Bob 以为这个密钥就是他和 Alice 所共享的密钥。

### 攻击 8.1 对 Diffie-Hellman 密钥交换协议的中间人攻击

共同输入: 同协议 8.1



- 1 Alice 选择  $a \in_U [1, p-1)$ , 计算  $g_a \leftarrow g^a(\bmod p)$ ;  
发送  $g_a$  给 Malice(“Bob”);
- 1' Malice(“Alice”)对某个  $m \in [1, p-1)$ , 计算  $g_m \leftarrow g^m(\bmod p)$ ;  
发送  $g_m$  给 Bob;
- 2 Bob 选择  $b \in_U [1, p-1)$ , 计算  $g_b \leftarrow g^b(\bmod p)$ ;  
发送  $g_b$  给 Malice(“Alice”);
- 2' Malice(“Bob”)向 Alice 发送:  $g_m$ ;
- 3 Alice 计算  $k_1 \leftarrow g_m^a(\bmod p)$ ; (\* 由于 Malice 能够计算  $k_1 \leftarrow g_m^a(\bmod p)$ , 这个密钥由 Alice 和 Malice 共享 \*)
- 4 Bob 计算  $k_2 \leftarrow g_m^b(\bmod p)$ 。(\* 由于 Malice 能够计算  $k_2 \leftarrow g_m^b(\bmod p)$ , 这个密钥由 Bob 和 Malice 共享 \*)

类似地, Malice 可以伪装成 Bob, 并同 Alice 协商另一个密钥  $g^{am}(\bmod p)$ 。以上两个过程完成之后, Malice 用这两个密钥就可以在 Alice 和 Bob 之间阅读或转发“保密”通信, 或者对于其中一方, 伪装另一方。

由于这个协议没有提供对协议消息源的认证服务, 对 Diffie-Hellman 密钥交换协议的中间人攻击是可能的。为了协商一个仅由 Alice 和 Bob 专门共享的密钥, 在协议的运行过程当中, 参与者必须确定收到的消息的确来自目标参与者。在第 11 章, 我们将研究认证技术; 那里(11.6 节), 我们还将介绍安全应用 Diffie-Hellman 密钥交换协议的方法。

## 8.4 Diffie-Hellman 问题和离散对数问题

Diffie-Hellman 密钥交换协议中共享密钥的保密性就是已知  $g_a$  和  $g_b$ , 计算  $g^{ab}(\bmod p)$  的问题。这个问题称为计算 **Diffie-Hellman 问题**(CDH 问题)

**定义 8.1 计算 Diffie-Hellman 问题 (CDH 问题) (在有限域中)**

输入  $\text{desc}(\mathbb{F}_q)$ : 对有限域  $\mathbb{F}_q$  的描述;  
 $g \in \mathbb{F}_q^*$ :  $\mathbb{F}_q^*$  的生成元;  
 $g^a, g^b \in \mathbb{F}_q^*$ , 其中整数  $0 < a, b < q$ 。

输出  $g^{ab}$ 。

我们在有限域  $\mathbb{F}_q$  中用公式表示了这个问题的一般形式。8.3 节中的 Diffie-Hellman 密钥交换协议运用了一种特殊的情况。为了把问题形式化, 在定义某个一般问题时, 例如, 一个假设等, 我们应该尽可能地考虑一般情况, 而对一个形式化定义做表面解释时, 我们经常会用一些特殊的例子, 这将有助于清楚地解释概念。

如果 CDH 问题是容易的, 则  $g^{ab} \pmod{p}$  可以由  $p, g, g^a, g^b$  来计算得到, 而这些参数是作为协议消息的一部分传送的。按照我们对攻击者能力的假设 (见 2.3 节), 攻击者能够得到这些数据。

进一步, CDH 问题基于离散对数问题 (DL 问题) 的困难性。

**定义 8.2 离散对数问题 (DL 问题) (在有限域中)**

输入  $\text{desc}(\mathbb{F}_q)$ : 有限域  $\mathbb{F}_q$  的描述;  
 $g \in \mathbb{F}_q^*$ :  $\mathbb{F}_q^*$  的一个生成元;  
 $h \in \mathbb{F}_q^*$ 。

输出 惟一的整数  $a < q$ , 满足  $h = g^a$ 。

我们用  $\log_g h$  表示整数  $a$ 。

DL 问题看起来与在有理数范围内求一般的对数类似, 但却并不相同。在有理数中我们只需要近似“解”, 而 DL 问题是定义在一个离散的域内, 在这里面的解是精确的。

第 4 章讨论的现代公钥密码体制的安全性理论建立在复杂性理论基础之上。在这个基础之上, 公钥密码体制的安全性是基于某些假设有条件安全的, 这些假设就是假设某些问题是困难的。CDH 问题和 DL 问题是两个假设为困难的问题。直观上, 我们可以立即看到这两个问题的困难性取决于问题的规模 (这里也就是有限域  $\mathbb{F}_q$  的大小), 以及参数的选择 (这里就是公开参数  $g$  和秘密数据  $a, b$ )。显然, 当参数选得比较小时, 这两个问题并不是困难的; 很快我们将会进一步看到, 若参数选得不好, 这两个问题也不是困难的。所以, 要精确描述困难性就必须准确地阐明问题的规模和参数的选择。利用第 4 章建立的复杂性理论基础, 我们现在可以精确地描述关于这两个问题困难性的假设。读者可以回顾第 4 章来温习将在以下的形式化论述中用到的几个概念 (例如“ $1^k$ ”、“概率多项式时间”以及“关于  $k$  可以忽略的量”)。

**假设 8.1 计算 Diffie-Hellman 假设 (CDH 假设)** 解决 CDH 问题的算法是一个 PPT 算法  $\mathcal{A}$ , 对于一个概率  $\epsilon > 0$ , 满足

$$\epsilon = \text{Prob}[(g^{ab} \leftarrow \mathcal{A}(\text{desc}(\mathbb{F}_q), g, g^a, g^b))]$$

其中  $\mathcal{A}$  的输入由定义 8.1 确定。

令  $\mathcal{IG}$  是一个实例生成器, 输入  $1^k$ , 在  $k$  的多项式时间内运行, 输出 (i)  $\text{desc}(\mathbb{F}_q)$ , 其中  $|q| = k$ , (ii) 一个生成元  $g \in \mathbb{F}_q^*$ 。

我们说 $IG$ 满足计算 Diffie-Hellman(CDH)假设,如果对于所有足够大的 $k$ ,对于 $k$ 不可忽略的概率 $\epsilon > 0$ ,不存在由 $IG(1^k)$ 所产生的 CDH 问题的求解算法。

**假设 8.2 离散对数假设(DL 假设)** 解决 DL 问题的算法是一个 PPT 算法 $A$ ,对于一个概率 $\epsilon > 0$ ,满足

$$\epsilon = \text{Prob}[\log_g h \leftarrow A(\text{desc}(\mathbb{F}_q), g, h)]$$

其中 $A$ 的输入由定义 8.2 确定。

令 $IG$ 是一个实例生成器,输入 $1^k$ ,在 $k$ 的多项式时间内运行,输出(i) $\text{desc}(\mathbb{F}_q)$ ,其中 $|q| = k$ ,  
(ii)一个生成元 $g \in \mathbb{F}_q^*$ , (iii) $h \in \mathbb{F}_q^*$ 。

我们说 $IG$ 满足离散对数(DL)假设,如果对所有足够大的 $k$ ,对于 $k$ 不可忽略的概率 $\epsilon > 0$ ,不存在由 $IG(1^k)$ 所产生的 DL 问题的求解算法。

简而言之,这两个假设说明了在充分大的有限域中,几乎对所有的情形,不存在求解 CDH 问题或 DL 问题的有效算法。一部分可忽略的例外是由于存在一些弱的特例。

但是,对这两个假设需要做更适宜而详尽的阐述。首先我们用“形式的语调”给出几个重要的注释。

#### 注释 8.1

1. 在假设 8.1 和 8.2 中,各自的概率空间应该考虑 (i) 事件空间,即任意的有限域和任意的样本元素(这一点的重要性将在 8.4.1 节中讨论), (ii) 有效算法当中随机运算的空间。之所以要考虑(ii)是因为我们引入的“多项式时间”或“有效”算法包含了随机算法(见 4.6 节的定义 4.6.6)。
2. 两个形式化问题中的参数 $k$ 称为**安全参数**。 $IG(1^k)$ 是这个域和元素的随机实例。根据在 4.4.6.1 节对概率素数生成的研究和 5.4 节中关于域的构造,我们知道 $IG(1^k)$ 确实在 $k$ 的多项式时间内结束。现在普遍认为,对于有限域中的 DLP,  $k = 1024$ 是安全参数设置的下界。这个下界是由求解有限域中 DLP 的亚指数时间算法(指数积分)得到的结果。亚指数复杂度表达式在式(8.4.2)中给出。对于 $|q| = 1024$ ,这个表达式得到的一个值大于 $2^{80}$ 。这就是为什么 $k = 1024$ 成了一个广泛认可的下界。因此,像在这两个假设中所约定的“对于足够大的 $k$ ”,我们应该仅考虑 $k$ 大于这个下界。
3. DL 假设成立意味着函数

$$g^x: \mathbb{Z}_q \mapsto \mathbb{F}_q^* \quad (8.4.1)$$

是单向的。所以,DL 假设成立意味着单向函数的存在。广泛认为 DL 假设应该成立(基于确信 $P \neq NP$ ,见 4.5 节),或者式(8.4.1)中的函数应该是单向的,或者换句话说,单向函数应该是存在的。

4. 目前还不知道式(8.4.1)中的函数是否是陷门函数(见 8.1 节性质 8.1 中单向陷门函数的含义)。也就是说,还不知道如何在这个函数中嵌入陷门信息来有效地求得这个函数的逆(即运用陷门信息由 $g^x$ 计算 $x$ 的一个有效方法)。但是,如果这个函数用一个合数模(这个函数仍然是一个单向函数),那么这个函数就成了一个陷门函数,其中模的素分解就是陷门信息。详细技术上的细节请参阅[231, 226, 230]。□

我们仍然需要用更“通俗的语言”来解释这两个假设。

这两个假设本质上是说“不存在求解这两个问题的  $k$  的多项式时间算法”。但是,我们必须很仔细地阅读这段话。“poly( $k$ )解法”,如果存在的话,运行时间为  $k^n$ ,  $n$  为某个整数。另一方面,我们知道存在一个 DLP 的“亚指数解法”运行时间为

$$\text{sub\_exp}(q) = \exp(c(\log q)^{\frac{1}{3}}(\log \log q)^{\frac{2}{3}}) \quad (8.4.2)$$

其中  $c$  是一个小的常数(例如,  $c < 2$ )。结合“不存在 poly( $k$ )解法”和“存在 sub\_exp( $q$ )解法”,本质上是说  $k^n$  远远小于 sub\_exp( $k \log 2$ ) (对于  $k = |q| = \log_2 q$ , 我们有  $\log q = k \log 2$ )。但是,这个“远远小于”关系只有当  $n$  固定和  $k$  ( $n$  的一个函数)足够大的时候才成立。现在我们就清楚地说明这一点。

假设  $k$  不是足够大。对  $\text{poly}(k) = k^n$  和  $\text{sub\_exp}(k \log 2)$  取自然对数,比较以下两个等式:

$$n \log k \text{ 和 } ck^{\frac{1}{3}} [\log(k \log 2)]^{\frac{2}{3}}$$

该比较可化简如

$$n (\log k)^{\frac{1}{3}} \text{ 和 } ck^{\frac{1}{3}}$$

的比较。现在我们看到,当  $n$  在  $ck^{\frac{1}{3}}$  的数量级上时,这个已知的亚指数算法比假定的“ $k$  的非多项式时间算法”快。“非 poly( $k$ )算法”的真实意思是把  $k$  看做是一个无界(所以如同在上面两个假设所陈述的那样可以“足够大”)的变量,其中  $n$  是固定的常数。实际上,  $k$  不能没有界。尤其对于普遍认可的安全参数的下界:  $k = 1024$ , 并且对于  $c < 2$ , 确实存在一个“poly( $k$ )解法”,其运行时间为  $k$  的 9 次多项式(通过习题 8.4 来证实这一点)。

到目前的讨论为止,我们得到了一个“非 poly( $k$ )解法”的渐进解释:  $k$  无界并且足够大。实际上  $k$  一定是有界的,所以  $k$  的多项式解法一定存在。然而,我们可以规定  $k$  的一个下界,使我们可以比较满意地使得多项式时间解法所运行的时间是一个难以应付的大数。实际上,普遍认为  $k = 1024$  时就能做到这一点。

“非多项式时间解法”的渐进意思适用于在本书中其余部分出现的所有基于复杂度理论的困难性假设。

最后,看一下这两个问题之间的关系。

注意如果能够得到  $a = \log_g g_1$  或  $b = \log_g g_2$ , 就可以计算

$$g^{ab} = g_1^b = g_2^a$$

即,由求解 DLP 的有效算法可以得到一个求解 CDH 问题的有效算法。所以,如果 DL 假设不成立,就不能有 CDH 假设。我们说 CDH 问题比 DL 问题弱,或等价地,CDH 假设是一个比 DL 假设更强的假设。该关系的逆是一个公开问题:

如果 CDH 假设是错误的,那么 DL 假设会正确吗?

Maurer 和 Wolf 给出了对这两个问题关系的一个很强的启发式阐述;他们指出,这两个问题很可能是等价的[192]。

#### 8.4.1 任意参数对于满足困难假设的重要性

我们应该强调在 DL 假设中要求的任一情形的重要性。考虑  $\mathbb{F}_q^*$  和由  $h \equiv g^a \pmod{p}$  求  $a$  的问题,其中  $p$  是  $k$  比特素数。



我们知道  $a$  是  $\mathbb{Z}_{p-1}$  中的一个元素。如果  $p-1 = q_1 q_2 \cdots q_\ell$  的每一个因子  $q_i$  都很小(对于  $i = 1, 2, \dots, \ell, q_i \leq \text{polynomial}(k)$ ), 则求解离散对数的问题可以转化为由  $h^{(p-1)/q_i} \pmod{p}$  求  $a_i \equiv a \pmod{q_i}$  的问题, 而  $a_i$  是小的整数, 可以在  $k$  的多项式时间内求得。在求得  $a_1, a_2, \dots, a_\ell$  之后, 应用中国剩余定理(定理 6.7)就可以求得  $a$ 。这个思想是基于当  $p-1$  无大素因子时, Pohlig 和 Hellman 求解模  $p$  的 DL 问题的多项式时间算法[233]。显然, 如果  $p-1$  的每一个素因子的大小都不超过  $k$  的多项式, 则 Pohlig-Hellman 算法有  $k$  的多项式运行时间。

一个素数  $p$ , 如果  $p-1$  不包含大的素因子, 则称为平滑素数。但有时我们也说“ $p-1$  是平滑的”, 它们所表达的意思是相同的。一个排除平滑素数情况的标准方法就是构造一个素数  $p$ , 满足  $p-1$  被另一个大素数  $p'$  整除。由定理 5.2(2), 循环群  $\mathbb{F}_p^*$  包含惟一一个  $p'$  阶子群。如果  $p'$  是公开的, Diffie-Hellman 密钥交换协议的用户可以确保该协议运行在一个大的子群当中; 它们所需要做的就是找到元素  $g \in \mathbb{F}_p^*$  满足

$$g^{(p-1)/p'} \not\equiv 1 \pmod{p}$$

这个元素  $g$  生成了阶为素数  $p'$  的子群。Diffie-Hellman 密钥交换协议应该把这样生成的参数  $(p, p', g)$  作为共同的输入。可以接受的素数  $p'$  的规模应该至少为 160 (二进制比特), 即  $p' > 2^{160}$  (10.4.8.1 节中的讨论中还会碰到)。

在一般的阶为大整数的有限交换群中, 如有限域的阶为大素数的子群, 或定义在有限域上的椭圆曲线上的点所组成的群(群的构造见 5.5 节; 椭圆曲线离散对数问题, 即 ECDLP 见 5.5.3 节), DLP 和 CDH 问题也被认为是困难的。于是, 在这些群中, Diffie-Hellman 密钥交换协议也能很好地运行。

当知道要求的离散对数比较小时, 存在几个有效的指数时间算法。我们已经描述了 Pollard 的  $\lambda$  方法(见 3.6.1 节)。求解小的离散对数问题在了很多密码协议中都很有用。

对 DLP 的研究是很活跃的。Odlyzko 对这一领域做了概述, 包含了关于这一问题的大量文献[223]。

## 8.5 RSA 密码体制(教科书式)

RSA 是最熟悉的公钥密码体制, 以它的发明者 Rivest、Shamir 和 Adleman 的名字命名[248]。RSA 是基于 Diffie 和 Hellman 所想像的单向陷门函数的定义[98, 99], 而给出的第一个公钥密码的实际实现。

在算法 8.1 中具体描述了 RSA 密码体制, 这是 RSA 加密的一个教科书式版本。

我们现在证明, 算法 8.1 描述的系统的的确是一个密码体制, 即 Alice 的解密过程将真实地得到与 Bob 加密的相同的明文。

### 算法 8.1 RSA 密码体制

#### 密钥建立

为了生成用户的基本参数, 用户 Alice 执行以下步骤:

1. 随机选择两个素数  $p$  和  $q$ , 满足  $|p| \approx |q|$ ; (\* 应用 Monte-Carlo 找素数的算法, 即算法 4.7 \*)



2. 计算  $N = pq$ ;
3. 计算  $\phi(N) = (p-1)(q-1)$ ;
4. 随机选择整数  $e < \phi(N)$ , 满足  $\gcd(e, \phi(N)) = 1$ , 并计算整数  $d$  满足

$$ed \equiv 1 \pmod{\phi(N)}$$

(\* 由于  $\gcd(e, \phi(N)) = 1$ , 这个同余式的确有一个解  $d$ , 可以应用扩展的欧几里得算法求得(算法 4.2) \*)

5. 公开她的公钥  $(N, e)$ , 安全地销毁  $p, q$  和  $\phi(N)$ , 并保留  $d$  作为她的私钥。

#### 加密

为了秘密地将  $m < N$  发送给 Alice, 发送者 Bob 生成密文  $c$  如下

$$c \leftarrow m^e \pmod{N}$$

(\* 虽然实际上明文空间是  $\mathbb{Z}_N^*$ , 在 Bob 看来, 明文空间仍然是小于  $N$  的所有正整数集合。\*)

#### 解密

为了解密密文  $c$ , Alice 计算

$$c \leftarrow m^d \pmod{N}$$

由模运算的定义(见 4.3.2.5 节中的定义 4.4), 算法 8.1 中的同余式  $ed \equiv 1 \pmod{\phi(N)}$  意味着存在某个整数  $k$ , 使得

$$ed = 1 + k\phi(N)$$

所以, 由 Alice 的解密过程得到的数是

$$c^d \equiv m^{ed} \equiv m^{1+k\phi(N)} \equiv m \cdot m^{k\phi(N)} \pmod{N} \quad (8.5.1)$$

我们应该注意, 对于  $m < N$ , 几乎总是有  $m \in \mathbb{Z}_N^*$  (与  $N$  互素的整数构成的乘法群)。事实上,  $m \notin \mathbb{Z}_N^*$  的情况就是存在某个正数  $u < q$  或  $v < p$ , 有  $m = up$  或者  $m = vq$ 。在这两种情况下, Bob 可以通过计算  $\gcd(m, N)$  来分解  $N$ 。假设整数分解是困难的(我们将阐明整数分解问题及其困难性假设), 我们可以假设 Bob 要加密的任意消息  $m < N$  满足  $m \in \mathbb{Z}_N^*$ 。

对于  $m \in \mathbb{Z}_N^*$ , 由拉格朗日定理(推论 5.2), 我们有

$$\text{ord}_N(m) \mid \# \mathbb{Z}_N^* = \phi(N)$$

对于所有的  $m \in \mathbb{Z}_N^*$ , 上式总成立。由群元素阶的定义(见 5.2.2 节的定义 5.9), 这就意味着对所有的  $m \in \mathbb{Z}_N^*$ ,

$$m^{\phi(N)} \equiv 1 \pmod{N}$$

显然, 对于任意整数  $k$ , 进一步得到

$$m^{k\phi(N)} \equiv (m^{\phi(N)})^k \equiv 1 \pmod{N}$$

因此, 式(8.5.1)中的值的确是  $m$ 。

**例 8.2** 令 Alice 选择  $N = 7 \times 13 = 91$  和  $e = 5$ , 则  $\phi(N) = 6 \times 12 = 72$ 。应用算法 8.2(输入  $(a, b) = (72, 5)$ ), Alice 得到:

$$72 \times (-2) + 5 \times 29 = 1$$

即  $5 \times 29 \equiv 1 \pmod{72}$ 。于是, Alice 计算得到的 29 作为她的秘密解密指数。公开  $(N, e) = (91, 5)$  作为 RSA 体制的公开参数。

令 Bob 加密明文  $m = 3$ , Bob 计算

$$c = 3^5 = 243 \equiv 61 \pmod{91}$$

得到的密文是 61。

为了解密密文 61, Alice 计算

$$61^{29} \equiv 3 \pmod{91}$$

□

## 8.6 公钥密码体制的分析

我们说“密码体制 X 对于攻击 Y 是安全的,但是对于攻击 Z 是不安全的”是有道理的,即密码体制的安全性是根据攻击来定义的。主动攻击通常有三种方式,这些主动攻击的方式将用于对本章其余部分所介绍的密码体制的分析。它们的定义如下。

### 定义 8.3 密码体制的主动攻击

**选择明文攻击(CPA)** 攻击者选择明文消息并得到加密服务,产生相应的密文。攻击者的任务是用所得到的明-密文对来降低目标密码体制的安全性。

**选择密文攻击(CCA)** 攻击者选择密文消息并得到解密服务,产生相应的明文。攻击者的任务是用所得到的明-密文对来降低目标密码体制的安全性。在解密服务停止后,即在得到目标密文之后,解密服务立即停止,如果攻击者能够从“目标密文”中得到保密明文的信息,则就说攻击是成功的。

**适用性选择密文攻击(CCA2)** 这是一个 CCA,而且除了对“目标密文”解密外,永远能够得到解密服务。

我们可以用以下情形来想像上述攻击类型:

- 在 CPA 中,攻击者有一个加密盒子。
- 在 CCA 中,攻击者可以有条件地使用解密盒子:在交给攻击者目标密文之前关闭解密盒子。
- 在 CCA2 中,在攻击者得到目标密文之前或之后,只要攻击者不把目标密文输入解密盒子(这个惟一的限制是合理的,否则攻击者就没有任何需要解决的困难问题了),他就可以一直使用这个解密盒子。

在所有的情况下,攻击者都不应该拥有相应的密钥。

CPA 和 CCA 原来是作为攻击对称密码系统所提出的主动密码分析模型,在对称密码系统中,攻击者的目标就是用他从攻击中得到的明-密文对减弱目标加密系统(可参阅文[286]的 1.2 节)。它们已经用于规范对公钥系统的主动攻击。我们应该指出以下有关公钥密码系统的三个点细节:

- 在公钥系统下,由于给定了公钥,任何人都可以完全控制加密算法,这样任何人都总是可以得到公钥系统的加密服务。换句话说,CPA 永远可以用来攻击公钥密码系统。于是,如果对公钥密码系统的一个攻击没有用到任何解密服务,我们就可以称这个攻击为 CPA。因此,显然任何一个公钥密码系统必须抵抗 CPA,否则它就不是一个有用的密码系统。
- 一般地,大多数公钥密码体制所基于的数学问题都有一些很好的代数结构性质,如闭包、结合律和同态等(回顾第 5 章中的代数性质)。一个攻击者可以运用这些很好的性质,并通过巧妙的计算组成一条密文。如果攻击者能得到解密服务,则他的巧妙计算可能使他得到一些明文信息,或者甚至是目标加密系统的私钥,否则要得到私钥对他来说在计算上是不可行的。所以,公钥系统特别容易受到 CCA 和 CCA2 的攻击。作为一个一般的准则,我们已经在性质 8.2(ii)中指出,私钥的拥有者应该注意不要为任何人提供解密服务。本章所介绍的每一个公钥系统都必须遵循这个建议。在第 14 章,我们将介绍更强的公钥系统,它们不需要用户一直保持这样的警惕。
- 看起来 CCA 限制太大了。在应用中,处于攻击下的用户(被要求提供解密服务)实际上未必知道攻击的存在。所以用户就不知道何时应该停止提供解密服务。我们一般假设普通用户不知道攻击者的存在,所以攻击者一直能够得到解密服务。另一方面,由于攻击者总能够自己来执行选择明文的加密“服务”,所以任何公钥系统都必须抵抗 CPA。由于这个原因,我们主要考虑抵抗 CCA2 的方法。

## 8.7 RSA 问题

抵抗 CPA, RSA 的安全性基于计算密文  $C$  模合数  $n$  的  $e$  次根的困难性。这就是所谓的 RSA 问题。

### 定义 8.4 RSA 问题

输入  $N = pq$ , 其中  $p$  和  $q$  是素数;  
 $e$ : 一个整数满足  $\gcd(e, (p-1)(q-1)) = 1$ ;  
 $c \in \mathbb{Z}_N^*$ 。

输出 惟一的整数  $m \in \mathbb{Z}_N^*$ , 满足  $m^e \equiv c \pmod{N}$ 。

所有公钥系统的安全性都基于困难问题,这一点没有什么区别,并假设 RSA 问题只有在选择适当的参数条件下才是困难的。

**假设 8.3 RSA 假设** RSA 算法是一个 PPT 算法  $\mathcal{A}$ , 对于一个概率  $\epsilon > 0$  满足:

$$\epsilon = \text{Prob}[m \leftarrow \mathcal{A}(N, e, m^e \pmod{N})]$$

其中  $\mathcal{A}$  的输入由定义 8.4 来定。

令  $\mathcal{IG}$  是一个 RSA 生成器, 输入  $1^k$ , 在  $k$  的多项式时间内输出 (i) 一个  $2k$  比特的模  $N = pq$ , 其中  $p$  和  $q$  是两个不同的均匀分布的素数, 长度均为  $k$  比特, (ii)  $e \in \mathbb{Z}_{(p-1)(q-1)}^*$ 。

我们说  $\mathcal{IG}$  满足 RSA 假设, 如果对于所有充分大的  $k$ , 对于  $k$  不可忽略的概率  $\epsilon > 0$ , 不存在由  $\mathcal{IG}(1^k)$  所产生的 RSA 问题的求解算法。

与注释 8.1(3)中的讨论相似(见 8.4 节),我们知道 RSA 假设成立意味着单向函数的存在。也与注释 8.1(4)有关系,单向函数意味着 RSA 假设是一个陷门函数:模的素分解使其存在一个有效的逆。

我们应该注意,在这个假设中的概率空间包括参数空间、明文消息空间以及求解 RSA 问题的随机化算法的一些随机计算。

我们进一步指出,在对 RSA 假设的描述当中,(声明的)算法把指数  $e$  当成输入的一部分。这就准确地阐明了问题的目的:已知加密指数的情况下,攻破 RSA 问题。RSA 问题还有另一种形式,称为**强 RSA 问题**([18,113,86]);它的目标是:对于某些奇数的加密指数  $e > 1$ ,可能是根据算法的选择,已知这样的  $e$ ,解 RSA 问题。显然,解强 RSA 问题比固定加密指数的 RSA 问题简单。普遍认为(假设)强 RSA 问题仍然是不可求解的。所以某些加密算法或者协议的安全性基于这个困难性(**强 RSA 假设**)。

显然,对于公开密钥  $(N, e)$ ,如果  $m < N^{1/e}$ ,则加密  $c = m^e \pmod{N}$  将不会用到模简约运算,所以通过在整数范围内求  $e$  次根可以很快地求得  $m$ 。这就是为什么应该避免  $e = 3$  这种情况。当  $e = 3$  时,如果同一条消息  $m$  用三个不同的模加密:  $c_i = m^3 \pmod{N_i}, i = 1, 2, 3$ ,则由于这三个模数是两两互素的,可以运用中国剩余定理算法(算法 6.1)来计算  $C = m^3 \pmod{N_1 N_2 N_3}$ 。由于  $m < (N_1 N_2 N_3)^{1/3}$ ,加密指数运算实际上与在整数上运算一样。于是对  $C$  的解密就是在求 3 次整数根,这可以有效地完成(见习题 8.8 的提示)。

Coppersmith[83]进一步把这种平凡的情况扩充到非平凡的情况:对于  $m' = m + t$ ,其中  $m$  是知道的,  $t$  是不知道的,但是  $t < N^{1/e}$ ,已知  $c = m'^e \pmod{N}$ ,可以有效地求得  $t$ 。因为在实际应用中,通常知道明文的部分信息(我们将在第 15 章看到这种情况),现在普遍认为 RSA 加密应该避免用很小的加密指数。广泛认可的加密指数是  $e = 2^{16} + 1 = 65\,537$ ,这也是个素数。这个指数提高了加密效率,同时也避免了小指数攻击。

如果解密指数  $d$  很小, RSA 也是 CPA 不安全的。当  $d < N^{1/4}$  时, Wiener 基于对  $e/N$  的连分式展开发现了一种求  $d$  的方法[300]。这个结果改进到  $d < N^{0.292}$ [51]。

## 8.8 整数分解问题

RSA 问题的困难性依赖于整数分解问题的困难性。

### 定义 8.5 整数分解问题(IF 问题)

输入  $N$ : 奇合数,至少有两个素因子。

输出 素数  $p$  满足  $p \mid N$ 。

再一次说明,只有在选择适当参数的条件下, IF 问题才是困难的。

**假设 8.4 整数分解假设(IF 假设)** 一个整数分解器是一个 PPT 算法  $\mathcal{A}$ , 满足概率  $\epsilon > 0$ :

$$\epsilon = \text{Prob}[\mathcal{A}(N) \text{ 整除 } N, 1 < \mathcal{A}(N) < N]$$

其中  $\mathcal{A}$  的输入由定义 8.5 确定。

令  $\mathcal{IG}$  是整数生成器,输入  $1^k$ , 在  $k$  的多项式时间输出  $2k$  比特的模  $N = pq$ 。其中  $p$  和  $q$  是  $k$  比特的随机奇素数

我们说  $IG$  满足整数分解 (IF) 假设, 如果对于所有足够大的  $k$ , 对于  $k$  不可忽略的概率  $\epsilon > 0$ , 不存在由  $IG(1^k)$  所产生的整数分解算法。

显然, 一个能解决 IF 问题的算法就能解决 RSA 问题, 因为 Alice 知道了大整数  $N$  的分解, 就能先计算出  $d \equiv e^{-1} \pmod{(p-1)(q-1)}$ , 从而可准确地对 RSA 密文解密。类似于 CDH 问题和 DL 问题间的关系, 此问题的逆也是一个公开问题: 如果 RSA 假设为伪, IF 假设能真吗?

与平滑素数使得 DL 问题弱化的情况类似,  $N$  的一个平滑素因子也会使得 IF 问题弱化。Pollard 用一个有效的分解算法, 即所谓的 **Pollard 的  $p-1$  算法** [239] 证明了这一点。Pollard 的  $p-1$  算法的想法可描述如下: 令  $p$  是  $N$  的一个素因子, 其中  $p-1$  的最大素因子不超过  $B = \text{Poly}(k)$ ,  $k = |N|$ ,  $\text{Poly}(k)$  是  $k$  的多项式 ( $B$  称为  $p-1$  的“平滑界”)。我们构造

$$A = \prod_{\text{primes } r < B} r^{\lfloor \log N / \log r \rfloor}$$

通过这个构造,  $p-1 \mid A$ , 由费马小定理 (定理 6.10), 对任意  $a$ , 满足  $\gcd(a, p) = 1$ , 则有  $a^A \equiv 1 \pmod{p}$ 。对于  $N$  的其他某个素因子  $q$ , 如果  $a \not\equiv 1 \pmod{q}$  (这是容易满足的), 则存在某个非  $q$  的倍数的整数  $\ell$ , 满足  $a^A - 1 \pmod{N} = \ell p$ 。于是,  $\gcd(a^A - 1 \pmod{N}, N)$  必为  $N$  的一个素因子, 并且如果  $N = pq$ , 它必是  $p$ 。只剩下证明  $A$  的大小是  $k$  的多项式, 这样计算  $a^A \pmod{N}$  需要  $k$  的多项式时间。

由素数定理 (可参阅文 [172] 的第 28 页), 小于  $B$  的素数至多有  $B/\log B$  个。于是, 我们有

$$A < B^{\lfloor \log N \rfloor \frac{B}{\log B}} < B^{k \frac{B}{\log B}}$$

即

$$|A| < kB \log 2 < k \text{Poly}(k)。$$

显然, 不等式的右端是  $k$  的多项式。于是,  $a^A \pmod{N}$  可以由  $k$  的多项式模  $N$  乘法计算得到 (采用算法 4.3)。注意, 不需要具体构造  $A$ ;  $a^A \pmod{N}$  可以通过对所有  $r < B$  的素因子, 计算  $a^{r^{\lfloor \log N / \log r \rfloor}} \pmod{N}$  得到。

可以很容易地构造 RSA 模  $N = pq$ , 满足  $p-1$  和  $q-1$  的平滑界为非 ( $|N|$  的) 多项式地小, 从而这个模就抗击这种分解方法。可以通过找大的素数  $p'$  满足  $p = 2p' + 1$  也是个素数, 大素数  $q'$  满足  $q = 2q' + 1$  也是个素数来找  $p$  和  $q$ 。这种形式的素数称为 **安全素数**, 两个安全素数相乘构成的 RSA 模称为 **安全 RSA 模**。在 RSA 密码系统中是否需要用安全 RSA 模是一个争论的话题。反对用 (可参阅文 [275]) 安全 RSA 模的人认为 RSA 模应该尽可能地随机, 况且对于随机选取的素数  $p$ ,  $p-1$  有大素因子的概率非常高。但是, 很多基于 IF 问题的协议确实需要用安全 RSA 模来达到由协议所递送的结果的正确性。

众所周知,  $N$  的素因子的部分信息可以产生一个分解  $N$  的有效算法。例如, 对于  $N = pq$ , 其中  $p$  和  $q$  的规模大致相同, 至多知道  $p$  的一半比特就可以在  $N$  的长度的多项式时间内分解  $N$ , 可参阅文 [83]。

如果不用任何关于输入合数的素因子的先验信息, 目前最好的分解算法是数域筛法 (NFS), 它的时间复杂度在式 (4.6.1) 中给出。于是, 与有限域上的 DLP 的安全参数情形类似, 为了能有较高的安全信心, 普遍认可的 RSA 模长度的下界设置是 1024 比特。

最近, 数域筛法展示了大规模并行计算的效果: 在 2002 年初, 世界范围内的 9000 个工作站联合运行一个并行算法, 这个算法运行了 4 个多月的时间, 分解了一个 512 比特 RSA 模 (RSA-512 挑战)。

对整数分解的研究是很活跃的,但不可能规划出一个确定的进步。Boneh 给出了关于 RSA 问题的概述[49]。有关 IF 问题这一领域进展的讨论和一些文献评述请参阅[200]的第3章。

## 8.9 教科书式 RSA 加密的不安全性

我们把算法 8.1 给出的 RSA 加密算法归为一种教科书式版本,这是因为这种版本的 RSA 加密算法在大多数教科书中都可见到。现在看一下教科书式 RSA 加密算法的安全(或不安全)性。

对于随机选择密钥和明文的情形,根据定义 8.4 和假设 8.3, RSA 有效的 CPA 攻击的存在意味着 RSA 假设一定是错误的。所以我们有

**定理 8.1** RSA 密码体制抗击 CPA 是“完全或无”安全的,当且仅当 RSA 假设成立。  $\square$

这里的“完全或无”安全的意思在性质 8.2(i)中有解释;而 CPA 表示攻击者像在 8.2(ii)中规定的那样是被动的。

但是,这个性质的保密性实际上不是很有用的,现在我们就来解释其中的原因。

### 8.9.1 中间相遇攻击和教科书式 RSA 上的主动攻击

我们首先考虑“完全或无”安全性。注意到这里的“完全”的意思是在一般情况下(消息与模的规模一样大)得到整个明文消息。在应用中不必一定是这样。在实际应用时,一条明文一般包含某些攻击者知道的非秘密的部分信息。教科书式 RSA 并没有隐藏关于明文的某些部分信息。例如,如果知道一条明文是一个小于 1 000 000 的数字(如一个秘密的支付或工资),则已知密文,攻击者可以用不超过 1 000 000 次的尝试加密找出明文。

通常,对于明文  $m(m < N)$ ,如果可用  $\sqrt{m}$  大小的存储器,则只需要  $\sqrt{m}$  次尝试就能以不可忽略的概率可以找出  $m$ 。这是 Boneh、Joux 和 Nguyen 给出的一个巧妙的发现[53],这个发现运用了分解小的明文消息不是困难的事实以及 RSA 函数的可乘性质。RSA 函数的可乘性质如下

$$(m_1 \times m_2)^e \equiv m_1^e \equiv m_2^e \equiv c_1 \times c_2 \pmod{N} \quad (8.9.1)$$

也就是说,明文的分解意味着相应的密文的分解。由于加密函数的混合变换性质几乎总是使得密文与模的长度相当,分解 RSA 密文通常是一个困难问题。但是,这个可乘的性质表明如果明文容易分解,则相应的密文也容易分解。后者分解容易导致了“中间相遇攻击”。这一点将在下面的例子中解释。

#### 攻击 8.2

条件:

令  $c = m^e \pmod{N}$ , 并且 Malice 知道  $m < 2^{\ell}$ 。  $m$  以不可忽略的概率是满足

$$m = m_1 \cdot m_2, m_1, m_2 < 2^{\frac{\ell}{2}} \quad (8.9.2)$$

的一个合数。

由 RSA 的可乘性质,我们有

$$c = m_1^e \cdot m_2^e \pmod{N} \quad (8.9.3)$$



Malice 执行如下步骤:

1. Malice 创建一个有序数据库

$$\{1^e, 2^e, 3^e, \dots, (2^{\frac{\ell}{2}})^e\} \pmod{N}$$

2. 接着他搜索这组有序数据, 尝试从这组数据中找到  $c/i^e \pmod{N}$  (其中  $i = 1, 2, \dots, 2^{\frac{\ell}{2}}$ ), 以找到

$$c/i^e \equiv j^e \pmod{N} \quad (8.9.4)$$

(\* 由式(8.9.2)和式(8.9.3), 用式(8.9.4)表示的一个“中间相遇攻击”将在  $2^{\frac{\ell}{2}}$  步计算  $i^e \pmod{N}$  之前出现。现在 Malice 知道了明文  $i, j$ , 他也就知道  $m = i \cdot j$  了。\*)

让我们度量一下 Malice 的代价。数据库的空间代价是  $2^{\frac{\ell}{2}} \cdot \log N$  比特。时间代价: 创建数据库中的元素的代价是  $O_B(2^{\frac{\ell}{2}} \cdot \log^3 N)$ , 排序数据的代价是  $O_B(\frac{\ell}{2} \cdot 2^{\frac{\ell}{2}})$ , 最后从排序的数据中找  $j^e \pmod{N}$  的代价是  $O_B(2^{\frac{\ell}{2}} \cdot (\frac{\ell}{2} + \log^3 N))$ 。最后一部分包括模指数运算和二分查找(用算法 4.4)的时间。于是, 用比特复杂度来度量的全部时间就是  $O_B(2^{\frac{\ell}{2}+1} \cdot (\frac{\ell}{2} + \log^3 N))$ 。如果能够提供  $2^{\frac{\ell}{2}} \cdot \log N$  比特空间, 则时间复杂度就会明显地小于  $2^\ell$ 。这个攻击的时间复杂度与平方根量级约化相当。

在明文消息的长度为 40 ~ 60 比特的情况下, 明文可以分解成两个大小相仿的整数的概率在 18% ~ 50% 之间(参阅文[53]的表 1)。

**例 8.3 攻击 8.2 的一个真实例子** 假设用 1024 比特的 RSA 按教科书式加密 DES 的一个 56 比特密钥。对于一个随机的 DES 密钥, 用  $2^{28} \cdot 1024 = 2^{38}$  比特存储(= 32GB)和  $2^{29}$  次模指数运算, 可以以不可忽略的概率找到这个密钥(把 DES 密钥分解成两个 28 比特的整数)。空间和时间代价实际上可以用一台好的个人计算机来处理, 而直接通过加密来搜索 DES 密钥需要  $2^{56}$  次模指数运算, 即使用一台专门的设备也是很昂贵的。□

现在我们知道, 一定不要用教科书式的 RSA 加密短的密钥或选择数小于  $2^{64}$  的口令。如果在应用中我们必须用 RSA 加密一些小的数, 甚至消息仅为一比特时会出现什么问题呢? 我们建议读者用将在第 15 章介绍的加密(包含一个基于 RSA 的方案)方法。

下一个例子进一步说明了教科书式 RSA 的 CPA 安全性是不充分的: 教科书式的 RSA 更不能抵抗主动攻击。

**例 8.4** 令 Malice 有条件地控制 Alice 的 RSA 解密盒子。这个条件是很“合理的”: 如果对 Malice 发送的密文的解密结果没有意义(看上去是随机的), 则 Alice 应该把这条明文返回给 Malice。我们说这个条件是“合理的”, 有下面两条理由:

- i) “对随机询问的随机应答”是很多密码协议中的一个相当标准的运行模式, 所以用户应该遵循“询问-应答”指令。事实上, 通常密码协议设计成允许协议参加者对解密盒子这样有条件地控制。例如, Needham-Schroeder 公钥认证协议(协议 2.5)就有这样的特性: Alice 被指令要求对 Bob 发送的密文解密。
- ii) 无论如何, 我们希望看起来随机的解密结果不应该向攻击者提供任何有用的信息。



现在假设 Malice 想知道他通过搭线窃听或从以前 Alice 和别人(不是和他)的秘密通信中截获的密文  $C \equiv m^e \pmod{N}$  所对应的明文。他选择一个随机数  $r \in_U \mathbb{Z}_N^*$ , 计算  $c' = r^e c \pmod{N}$ , 把他选择的密文  $c'$  发送给 Alice。Alice 解密后的结果是

$$c'^d \equiv rm \pmod{N}$$

由于  $r$  是  $\mathbb{Z}_N^*$  上的一个置换, 这个结果对于 Alice 来说是完全随机的。于是 Alice 把  $rm$  返回给 Malice。注意, Malice 知道  $r$ , 所以他计算模  $N$  的一次除法就能够得到  $m$ 。□

例 8.2 至例 8.4 表明, 教科书式 RSA 太弱了, 不适于实际应用。有必要对这些缺点做系统的修补。我们将分两步进行修补工作:

- 在第 14 章, 我们将强化公钥加密体制的安全性定义, 使其适于应用。
- 在第 15 章, 我们将研究适于应用的 RSA 加密方式, 这也是 RSA 加密的一个标准; 我们将在适于应用的强安全性定义基础上给出其安全性的形式证明。

## 8.10 Rabin 加密体制(教科书式)

基于计算模一个合数的平方根的困难性, Rabin 提出了一种公钥体制[242]。Rabin 的工作有很重要的理论价值; 它是第一个可证明安全的公钥体制: Rabin 密码的安全性恰好是 IF 问题的困难性(回忆我们对 RSA 情形的讨论: 不知道 RSA 问题是否等价于 IF 问题)。Rabin 密码体制的加密算法的效率非常高, 所以很适合某些特定的应用, 如由便携装置运行的加密。

算法 8.2 具体给出了 Rabin 密码体制, 我们注意到这是一个教科书式的 Rabin 加密。

### 算法 8.2 Rabin 密码体制

#### 密钥创建

为了生成用户的主要参数, 用户 Alice 执行以下步骤:

1. 选择两个随机素数  $p$  和  $q$ , 满足  $|p| \approx |q|$ ; (\* 与算法 6.1 中 RSA 模的生成相同 \*)
2. 计算  $N = pq$ ;
3. 随机选择一个整数  $b \in_U \mathbb{Z}_N^*$ ;
4. 公开她的公钥  $(N, b)$ , 把  $(p, q)$  作为她的私钥保密。

#### 加密

为了发送给 Alice 秘密消息  $m \in_U \mathbb{Z}_N^*$ , 发送者 Bob 按下面的方式生成密文  $c$ :

$$c \leftarrow m(m + b) \pmod{N}$$

#### 解密

为了解密密文  $c$ , Alice 对  $m < N$  求解二次方程

$$m^2 + bm - c \equiv 0 \pmod{N}$$

我们现在证明算法 8.2 描述的体制的确是一个密码体制, 即 Alice 的解密过程得到的明文确实与 Bob 加密的相同。

由初等数学知识我们知道, 这个方程的一般解可以写成

$$m \equiv \frac{-b \sqrt{\Delta_c}}{2} \pmod{N} \quad (8.10.1)$$

其中,

$$\Delta_c \stackrel{\text{def}}{=} b^2 + 4c \pmod{N} \quad (8.10.2)$$

由于  $c$  是由  $m \in \mathbb{Z}_N^*$  构成的,所以下面的二次方程

$$m^2 + bm - c \equiv 0 \pmod{N}$$

在  $\mathbb{Z}_N^*$  中有解,这些解包含由 Bob 发送的  $m$ 。这就意味着  $\Delta_c$  必为模  $N$  的二次剩余,即  $\text{QR}_N$  中的一个元素。

解密运算包括求模  $N$  的平方根运算。由 6.6.2 节所研究的平方根问题,我们知道求平方根问题的困难性等价于分解  $N$  的困难性(见推论 6.3)。由于只有 Alice 知道  $N$  的分解,所以只有她能够计算式(8.10.1)。运用算法 6.5, Alice 可以计算  $\sqrt{\Delta_c}$ 。在 6.6.2 节我们还知道,对于 Bob 发送的每一条密文都有 4 个不同的  $\sqrt{\Delta_c}$  值,所以有 4 个不同的解密结果。在应用中我们假设,明文消息应当包含冗余信息,使得 Alice 从 4 个解密的结果中能辨认正确的明文。10.4.3 节将给出“可识别冗余”的含义和一个把消息转化成包含可识别冗余的通用方法。

我们注意到如果  $N$  是所谓的 **Blum 整数**,即  $N = pq$ ,其中  $p \equiv q \equiv 3 \pmod{4}$ ,则计算模  $N$  的平方根就很容易(用算法 6.3 中  $p \equiv 3, 7 \pmod{8}$  时的情形,分别计算模  $p$  和  $q$  的平方根,接着运用中国剩余定理计算出这个平方根)。所以,实际上 Rabin 密码体制中的公开模数都选用 Blum 整数。

Rabin 加密算法仅包含一个乘法和一个加法运算,所以比 RSA 加密快很多。

**例 8.5** 令 Alice 选定  $N = 11 \times 19 = 209$  和  $b = 183$ 。作为 Rabin 密码体制的主要公开参数, Alice 公开  $(N, b) = (209, 183)$ 。

令 Bob 加密消息  $m = 31$ , Bob 运行 Rabin 加密:

$$c = 31 \times (31 + 183) \equiv 155 \pmod{209}$$

相应的密文是 155。

为了解密密文 155, Alice 首先运用式(8.10.2)计算  $\Delta_c$ :

$$\Delta_c = b^2 + 4c = 183^2 + 4 \times 155 \equiv 42 \pmod{209}$$

现在运用算法 6.5, Alice 求出 42 模 209 的 4 个平方根是 135, 173, 36, 74。最后她可以运用式(8.10.1)得到 4 个解密结果 185, 204, 31, 50。在 Rabin 密码体制的实际应用中,明文应该包含额外的信息使得接收者能找出正确的解密结果。□

## 8.11 教科书式 Rabin 加密的不安全性

我们有对教科书式 Rabin 密码体制更具破坏性的主动攻击。下面的定理以“可证明”的方式指出这种攻击。

## 定理 8.2

- I) Rabin 密码体制是抵抗 CPA 可证明“完全或无”安全的,当且仅当 IF 是困难问题。  
 II) 在 CCA 攻击下,Rabin 密码体制是根本不安全的。

**证明** (I)由于给出的 Rabin 密码体制解密过程用的是 RSA 模的分解,所以 Rabin 密码体制加密的安全性意味着分解 RSA 模的困难性。于是,对于(I)我们只需要证明另一方向命题:IF 问题的困难性意味着 Rabin 密码体制的安全性。

假设存在一个预言机  $\mathcal{O}$  能以不可忽略的概率  $\epsilon$  攻破 Rabin 密码体制,即

$$\text{Prob}\left[\mathcal{O}(c, N) = \frac{-b + \sqrt{\Delta_c}}{2} \pmod{N} \mid c \in_U \mathbb{Z}_N^*\right] \geq \epsilon$$

我们选择一条随机的消息  $m$ , 计算  $c = m(m+b) \pmod{N}$ , 并询问  $\mathcal{O}(c, N)$ , 将会以概率  $\epsilon$  返回  $m' \equiv \frac{-b + \sqrt{\Delta_c}}{2} \pmod{N}$ 。这里用  $\sqrt{\Delta_c}'$  表示  $\Delta_c$  的 4 个平方根中的任意一个。由定理 6.17(见 6.6.2 节), 我们知道以 1/2 的概率有

$$m' + \frac{b}{2} \equiv \frac{\sqrt{\Delta_c}'}{2} \not\equiv \pm \frac{\sqrt{\Delta_c}}{2} \equiv \pm \left(m + \frac{b}{2}\right) \pmod{N}$$

但是, 因为

$$\left(m' + \frac{b}{2}\right)^2 \equiv \frac{\Delta_c}{4} \equiv \left[\pm \left(m + \frac{b}{2}\right)\right]^2 \pmod{N}$$

如同定理 6.17 有

$$\gcd\left(m' + \frac{b}{2} \pm \left(m + \frac{b}{2}\right), N\right) = p \text{ 或 } q \quad (8.11.1)$$

即可以以不可忽略的概率  $\epsilon/2$  分解  $N$ 。这一点与分解 RSA 模的困难假设相矛盾(IF 假设)。这样我们就证明了(I)。

如果攻击者能够得到解密服务, 则陈述(II)显然成立: 解密服务恰好起到了在证明陈述(I)的过程中用到的预言机的作用! 由于攻击者生成(选择)要求解密预言机解密的密文, 这种攻击就是 CCA。□

定理 8.2 告诉我们两件事。第一, 从性质 8.2 (I)“完全或无”的意义上讲, 关于整数分解的困难性, Rabin 密码体制是可证明安全的(N.B. 给出的明文本身就是“完全或无”保密的, 即不知道明文的先验信息)。由于它使得(教科书式)Rabin 密码体制的加密安全性相关于一个著名的难题, 这是一个很强的、理想的结果。如果 IF 问题确实是困难的, 则在(I)的证明中所声称的预言机就不存在。但是, 我们应该特别注意为满足 CPA 安全性质修改的“完全或无”。这里的“完全”意味着在一般情况下(明文与模有相同的长度)求得整条明文消息。显然, 由于 Rabin 加密是确定的, 求得某些特殊的消息, 如一些短的消息, 就不像分解那样困难。在本节的最后, 在讨论对 Rabin 体制的中间相遇攻击时, 我们将回到这一点上来。

第二, 显然, 在 Rabin 密码体制中, 永远不要使自己成为一个解密预言机。CCA 完全攻破 Rabin 密码体制: 这个攻击的结果并不是仅仅求得了某些明文信息(如例 8.5 给出的 CCA2 对 RSA 密码体制的攻击), 它发现的是密钥拥有者的私钥, 所以攻击者将能够阅读所有用目标公钥加密的秘密消息。

例 8.6 在例 8.5 中,我们已经看到对于 Rabin 密码体制的主要公开参数  $(N, b) = (209, 183)$ , 密文 31 的 4 个解密结果是 185, 204, 31 和 50。

如果非私钥的拥有者得到这些数,即通过一个 CCA 攻击,他们可以用来分解模数 209。例如,运用式(8.11.1):

$$\gcd(204 - 185, 209) = 19$$

或者

$$\gcd((31 + 183/2) + (50 + 183/2), 209) = \gcd(264, 209) = 11 \quad \square$$

虽然我们已经提醒过, Rabin 加密体制私钥的拥有者永远不要提供解密服务,但在实际的应用中要求用户保持这样的高度警惕是不切实际的。因此,教科书式 Rabin 加密体制不适用于实际应用。在第 15 章,我们将介绍适于实际应用的 Rabin(与 RSA)体制的加密方法,也将给出这些加密体制适于应用的安全性的形式化证明。

我们应该注意到,由于 Rabin 密码体制的模与 RSA 密码体制的模相同,我们已经讨论的正确选取 RSA 模的方法也适应于 Rabin 模。

最后,中间相遇攻击也适应于下面教科书式 Rabin 加密体制的变形:

**加密**  $c = m^2 \pmod{N}$ 。

**解密** 计算  $c$  模  $N$  的平方根。

与教科书式 RSA 加密相似,在 Rabin 加密体制中容易分解小的明文消息和可乘的性质(见 8.9 节),使得中间相遇攻击能够成功,就像我们在例 8.3 中对教科书式 RSA 情况所演示的那样。

## 8.12 ElGamal 密码体制(教科书式)

ElGamal 设计了一个巧妙的密码体制[103]。这个密码体制是 Diffie-Hellman 单向陷门函数的一个成功应用,把函数转化成公钥加密体制。ElGamal 的工作在研究和应用领域激起了很强的兴趣,并持续到今日。我们在第 13 章(基于身份的 ElGamal 加密体制)和第 15 章(具有可证明强安全性的一个变形)将会看到这种密码体制的两个进一步发展。

出现继续进行 ElGamal 工作的强劲势头的原因之一就是,它使我们能将广泛认为可以依赖的困难问题用于解决公钥密码体制的安全性,CDH 问题被广泛认为同 DL 问题一样困难,后者被认为可与另一个问题匹敌,而该问题被广泛认为是可以依赖的困难问题:IF 问题(RSA 和 Rabin 的基础)。

ElGamal 密码体制在算法 8.3 中描述。注意,这是一个 ElGamal 的教科书式加密。

### 算法 8.3 ElGamal 密码体制

#### 密钥创建

为了创建用户的密钥数据, Alice 执行下列步骤:

1. 随机选择素数  $p$ ;
2. 计算  $\mathbb{F}_p^*$  的一个随机乘法生成元  $g$ ;
3. 随机选取  $x \in {}_U\mathbb{Z}_{p-1}$  作为她的私钥;

## 4. 计算她的公钥

$$y \leftarrow g^x \pmod{p}$$

5. 把  $(p, g, y)$  作为她的公开密钥公开, 把  $x$  作为她的私钥保存; (\* 与 Diffie-Hellman 密钥交换协议类似, 系统内的所有用户共享通用公开参数  $(p, q)$ 。\*)

## 加密

为了将消息  $m < p$  秘密地发送给 Alice, 发送者 Bob 选取  $k \in_U \mathbb{Z}_{p-1}$ , 按照下列运算计算密文对  $(c_1, c_2)$ :

$$\begin{cases} c_1 \leftarrow g^k \pmod{p} \\ c_2 \leftarrow y^k m \pmod{p} \end{cases} \quad (8.12.1)$$

## 解密

为了解密  $(c_1, c_2)$ , Alice 计算

$$m \leftarrow c_2 / c_1^x \pmod{p} \quad (8.12.2)$$

我们现在证明算法 8.3 的确是一个密码体制, 即 Alice 的解密过程得到的明文确实与 Bob 加密的相同。

由于

$$c_1^x \equiv (g^k)^x \equiv (g^x)^k \equiv y^k \equiv c_2 / m \pmod{p}$$

解密运算 (8.12.2) 的确恢复出明文  $m$ 。

解密步骤 (8.12.2) 中的除法运算需要运用扩展的欧几里得算法 (见算法 4.2), 通常这个算法比乘法的代价高。但是, Alice 可以通过计算

$$m \leftarrow c_2 c_1^{-x} \pmod{p}$$

来避免除法运算。

可以证明, 上面这种解密方法是可行的, 但是要注意这里的  $-x$  表示  $p-1-x$ 。

**例 8.7** 由例 8.1, 我们知道 3 是模 43 的本原根。令 Alice 选取 7 作为她的私钥。她计算自己的公钥

$$37 \equiv 3^7 \pmod{43}$$

Alice 公布她的公开密钥数据  $(p, g, y) = (43, 3, 37)$ 。

令 Bob 加密明文消息  $m = 14$ 。Bob 随机选择指数 26, 并计算

$$c_1 = 15 \equiv 3^{26} \pmod{43}, c_2 = 31 \equiv 37^{26} \times 14 \pmod{43}$$

得到的密文消息对是  $(15, 31)$ 。

为了解密密文消息  $(15, 31)$ , Alice 计算

$$14 = 31 / 36 \equiv 31 / 15^7 \pmod{43}$$

除法运算需要用算法 4.2, 但是 Alice 可以通过计算

$$14 = 31 \times 15^{42-7} \equiv 31 \times 6 \pmod{43}$$

来避免除法运算。

□

### 8.13 教科书式 ElGamal 加密的不安全性

ElGamal 密码体制的加密算法(8.12.1)是概率算法,它运用了一个随机的输入  $k \in_U \mathbb{Z}_{p-1}$ 。假设 Alice 的私钥  $x$  与  $p-1$  互素,则由定理 5.2(3)(见 5.2.3 节),她的公钥  $y \equiv g^x \pmod{p}$  仍然是  $\mathbb{F}_p^*$  的一个生成元(由于  $g$  是生成元),所以当  $k$  遍历  $\mathbb{Z}_{p-1}$  时,  $y^k \pmod{p}$  遍历  $\mathbb{F}_p^*$ 。由于模  $p$  的乘法运算是  $\mathbb{F}_p^*$  上的一个置换,对于任意的明文消息  $m \in \mathbb{F}_p^*$ ,  $k$  遍历  $\mathbb{Z}_{p-1}$  时,  $c_2 \equiv y^k m \pmod{p}$  遍历  $\mathbb{F}_p^*$  (见 6.2.2 节的定理 6.6)。因此,对于  $k \in_U \mathbb{Z}_{p-1}$ , 我们有  $c_2 \in_U \mathbb{F}_p^*$ 。这就意味着 ElGamal 加密实现了把明文消息均匀地分布到整个消息空间中。对于加密算法来说,这是理想的语意特性。

但是,我们不应过于乐观! ElGamal 加密不只是一组  $c_2$ ,而是一对  $(c_1, c_2)$ ,这两组是统计相关的。因此,与其他所有的公钥密码体制一样,ElGamal 密码体制的安全性有条件地基于一个困难问题的假设。而且,我们不久将看到(见 8.13.1 节),为了保持理想的语意特性,明文消息必须在群  $\langle g \rangle$  中。遗憾的是,通常在实际应用中不是这种情况。

首先,我们介绍关于 ElGamal 加密体制的“完全或无”安全性的结果。

**定理 8.3** 对于均匀分布在明文空间中的一条明文消息,ElGamal 加密体制抵抗 CPA 是“完全或无”安全的,当且仅当 CDH 问题是困难的。

**证明** ( $\Rightarrow$ ) 我们需要证明,如果 ElGamal 密码体制是安全的,则 CDH 假设成立。

假设相反,CDH 假设不成立。于是已知任意基于公钥  $y \equiv g^x \pmod{p}$  创建的密文  $(c_1, c_2) \equiv (g^k, y^k m) \pmod{p}$ , 一个 CDH 预言机将由  $(p, g, g^x, g^k)$  以不可忽略的概率计算出  $g^{xk} \equiv y^k \pmod{p}$ 。则可以以相同的概率求得  $m \leftarrow c_2 / y^k \pmod{p}$ 。这与 ElGamal 的安全性假设相矛盾。

( $\Leftarrow$ ) 我们需要证明,如果 CDH 假设成立,则不存在有效的算法,能以不可忽略的概率从 ElGamal 加密的密文恢复出明文消息。

假设相反,存在一个有效的预言机  $\mathcal{O}$  攻击 ElGamal 密码体制,即已知公钥  $(p, g, y)$  和密文  $(c_1, c_2)$ ,  $\mathcal{O}$  以不可忽略的概率  $\delta$  输出

$$m \leftarrow \mathcal{O}(p, g, y, c_1, c_2)$$

$m$  满足

$$c_2 / m \equiv g^{(\log_g y \log_g c_1)} \pmod{p}$$

对于任意的 CDH 问题实例  $(p, g, g_1, g_2)$ , 我们令  $(p, g, g_1)$  是公钥,令  $(g_2, c_2)$  是密文对,随机数  $c_2 \in \mathbb{F}_p^*$ 。则  $\mathcal{O}$  以概率  $\delta$  输出

$$m \leftarrow \mathcal{O}(p, g, g_1, g_2, c_2)$$

其中  $m$  满足

$$c_2 / m \equiv g^{(\log_g g_1 \log_g g_2)} \pmod{p}$$

这与 CDH 假设矛盾。 □

由于 ElGamal 密码体制的 CPA 安全性等价于 CDH 问题,我们关于 CDH 问题和 DL 问题(见 8.4 节)的讨论,如果仔细考虑公钥参数的选取,就都适用于 ElGamal 密码体制。如同

Diffie-Hellman 的密钥交换协议, ElGamal 密码体制也可在  $\mathbb{F}_q$  的一个大的素阶子群或者有限域上定义的椭圆曲线的点所组成的一个大群中运行。

### 8.13.1 教科书式 ElGamal 加密的中间相遇攻击和主动攻击

我们之所以把算法 8.3 的 ElGamal 密码体制归为教科书式体制,是因为它是一个很弱的加密体制。现在让我们来看究竟为什么。

ElGamal 加密体制在应用中常采用的形式甚至可能向一个被动的攻击者泄漏部分信息。实际上, ElGamal 密码体制一般采用阶为  $r = \text{ord}_p(g) \ll p$  的  $g$  来提高效率。在这种情况下, 如果一条消息  $m$  不属于  $\langle g \rangle$ , 则与对 RSA 攻击类似的中间相遇攻击(见例 8.3)也可以用来攻击教科书式 ElGamal 体制。这是因为, 对于密文  $(c_1, c_2) = (g^k, y^k m) \pmod{p}$ , Malice 能够得到

$$c_2' \equiv m' \pmod{p}$$

即 Malice 把 ElGamal 的“概率”加密体制转化成一种确定性的! 而且, 与教科书式 RSA 一样具有可乘性(在 8.9 节中解释)。因此, 对于一条容易分解的小消息, 与教科书式 RSA 的攻击完全一样, Malice 能够对  $m' \pmod{p}$  进行中间相遇攻击(文[53]发现了对 ElGamal 加密体制的这种中间相遇攻击)。

由这种攻击我们知道, 当一条明文消息不属于由  $g$  生成的子群的时候, ElGamal 密码体制就成为一种确定性的体制。由于确定的加密体制允许使用“尝试-错误”的方法来找小的明文消息, 显然它泄漏了部分信息, 例如秘密投标和工资数据。

最后, 我们给出一个对于主动攻击 ElGamal 是脆弱的例子。

**例 8.8** 令 Malice 可以有条件地控制 Alice 的 ElGamal 解密盒。如在例 8.5 中, 这个条件是“合理”的, 这是因为如果对 Malice 发送的密文的解密是一条无意义的消息(看起来是随机的), 则 Alice 需要把这个解密结果返回给 Malice。

令 Malice 通过搭线窃听或从 Alice 与别的什么人(不是与 Malice)之间秘密通信得到密文  $(c_1, c_2) \equiv (g^k, y^k m) \pmod{p}$ 。如果 Malice 想知道相应的明文, 他选择一个随机数  $r \in_U \mathbb{F}_p^*$ , 计算  $c_2' = rc_2 \pmod{p}$ , 并发送他选择的密文  $(c_1, c_2')$  给 Alice。由 Alice 解密后的结果将是

$$rm \pmod{p}$$

由于  $r < p$  的乘法运算是  $\mathbb{F}_p^*$  上的置换, 在 Alice 看来, 解密结果是完全随机的。于是, Alice 把这个结果  $rm$  返回给 Malice。Malice 拥有  $r$ , 于是他用一次模  $p$  的除法就可以得到  $m$ 。□

## 8.14 公钥密码系统需要更强的安全定义

我们已经介绍了几个基本的教科书式公钥密码体制。这些基本的体制可以看做是不同的单向陷门函数的直接应用(单向陷门函数的含义已由性质 8.1 给出)。

现在是给出这些教科书式密码体制的不安全特征总结的时候了。我们应该对教科书式公钥密码体制弱点的两个方面给出简短的讨论。

第一, 像性质 8.2 (i) 所陈述的那样, 在这一章中我们仅考虑了安全性一个很弱的定义: 在“完全或无”意义上的保密。在大多数公钥密码的应用中, 这样弱的保密定义远远不能达到满



意的程度,没有很大的用处。在很多的应用中,明文消息包含了攻击者知道的先验信息。例如,如果一种密码加密一张选票,则先验信息可能是“是”或“非”,或者是几个候选人的名字;于是,不管陷门函数多么强,攻击者只需要几次尝试性检验就可以确定正确的明文。在某些应用中,有关明文的一些先验信息会为攻击者提供一种无权享有的优势(我们将在 14.3.2 节中看到这样的攻击)。通常,教科书式加密算法不能很好地隐藏这部分信息。于是,需要更强的密码体制来安全地隐藏有关明文的任何先验信息。

第二,像性质 8.2 (ii) 所陈述的那样,在这一章中我们仅考虑了一个很弱的攻击方式:“被动攻击”。然而,对于本章介绍的教科书式密码体制,我们也演示了对它的主动攻击(例 8.5、8.7、8.9)。在这样的攻击中,攻击者可以准备一条巧妙计算的密文,并把它交给拥有密钥的用户,得到 CCA 或 CCA2 模式下的一种预言机解密服务。虽然作为一般的原则我们已经建议用户要抵抗主动攻击:私钥的拥有者应该时刻警惕不要提供解密服务,但是,考虑到要求普通用户也要时刻保持警惕状态是不实际的,建议用户不要对解密请求做应答并非是抵抗主动攻击的一个正确策略。

许多作者提出了采用涉及这两个方面更强的安全性定义的公钥密码体制。在第 14 章,我们将研究建立各种更强的保密性定义的课题以及如何实现形式化可证明安全。在第 15 章,我们将介绍适于应用的公钥密码体制,它们在很强的安全定义下是可证明安全的。

## 8.15 非对称密码与对称密码的组合

公钥密码学很好地解决了密钥分配问题。然而,在通常情况下,公钥密码函数运行在很大的代数结构之中,这就意味着昂贵的代数计算。相比较而言,对称密码函数一般更加有效。例如就 AES 而言,它在 256 元素的范围中运算;基本的运算如乘法和求逆可以通过“查表”法(回顾 7.7.4 节)实施,效率非常高。通常,公钥密码系统要比相应的对称密码系统所需的计算量大得多。

在应用中,尤其当需要加密大量的数据时,目前一种标准的方法是采用混合体制。在这样的体制中,公钥密码系统用来加密一个用于对称密码加密的所谓短期密钥;这就在发送者和接收者之间建立了共享的短期密钥;在这个共享的短期密钥控制下,采用对称密码系统对大量数据进行加密。这种组合方案发挥了这两种密码系统的优势:公钥系统易于密钥分配和对称密码系统的高效率。

在密码协议中,一个广泛应用的公钥与对称密码系统的组合就是所谓的数字信封技术。这是 RSA 密码体制与对称密码体制(如 DES、三重 DES 或 AES)的组合。这个通用的组合(RSA + DES 或 RSA + 三重 DES)是安全套接字层(SSL)协议的基本模式([138],我们将在第 12 章介绍 SSL 协议),这个协议已经被用于通用的 Web 浏览器上,如 Netscape、Internet Explorer 和 Web 服务器。在 SSL 协议中,协议的发起者(令为 Alice,通常处于 Web 客户的位置)首先下载通信另一方(令为 Bob,通常处于 Web 服务器的位置)的公钥数据;接下来 Alice(实际上,她的 Web 浏览软件)将生成一个随机会话密钥,用 Bob 的公钥加密(“装入信封”)这个会话密钥,并把这个“信封”发送给 Bob。在 Bob(实际上,他的 Web 服务器软件)解密这个“信封”并恢复这个会话密钥之后,双方现在就可以用这个会话密钥来加密他们随后的秘密通信。

在这个协议中,简单的混合加密方案在概念上很简单。但是它有两个缺陷。第一,这个方案所用的会话密钥是由一方(消息的发送者或是协议的发起者)生成的;另一方(消息的接收者

或协议的回应者)不得不完全依赖于发送者或协议发起者为安全通信而生成密钥的能力(或诚实)。这在某些环境下可能不是理想的,例如,在SSL协议的客户端-服务器环境下,这里的客户端就是发送者并且是用软件实现的,它在生成随机数方面的名声很差。

这个简单混合加密方案的第二个缺陷是非瞬息性。在混合加密体制中,能够强迫接收者出示她/他的私钥的搭线窃听者,就能够恢复所有的有效信息。这个缺点称为缺乏“前向保密性”。前向保密性意味着无论通过分析还是强迫,搭线窃听者都不可能由以前发送的密文在将来的时间恢复出明文消息。

如果混合加密方案的公钥密码部分采用 Diffie-Hellman 密钥交换协议,这两个缺点就能够克服。

首先来看如果混合方案采用 Diffie-Hellman 密钥交换协议时第一个缺点是如何消失的。在 Alice 和 Bob 双方运行的 Diffie-Hellman 密钥交换协议中,共享的秘密  $g^{ab}$  包含双方的随机输入: Alice 的贡献是  $a$ , Bob 的是  $b$ 。假设  $g$  生成了一个素阶群,并且满足  $g^a \neq 1$  和  $g^b \neq 1$ (见我们在 8.3 节中给出的“注意细节”), Alice(Bob)能够确信只要她(他)用了一个随机的整数,从  $g^{ab}$  推导的共享秘密会话密钥就是随机的。这是因为映射  $g^b \mapsto (g^b)^a$  和  $g^a \mapsto (g^a)^b$  在问题中是群的一个置换,所以均匀分布的指数(小于群的阶)把  $g^a(g^b)$  映射为一个均匀分布的群元素  $g^{ab}$ 。

其次,我们来看第二个缺陷是怎样克服的。注意到如果 Alice 和 Bob 像在 8.3 节中我们建议的那样谨慎地运行密钥交换协议,并且他们也正确地执行随后的通信,用 Diffie-Hellman 密钥交换协议的混合加密方案就具有前向保密性。为了谨慎地运行 Diffie-Hellman 密钥交换协议, Alice 和 Bob 应该交换他们的会话密钥,并在协议完成后立即销毁  $a$  和  $b$ 。为了正确进行以后的会话通信, Alice 和 Bob 应该在会话结束后销毁他们的会话密钥,并适当地处理他们所通信的明文消息。如果他们遵循这种相当标准的程序,显然胁迫手段也不会使搭线窃听者得到 Alice 和 Bob 所通信的明文消息。由 CDH 问题的困难性(见 8.4 节),此协议具有前向保密性,搭线窃听者的分析也不会成功。

最后我们指出,在很强的保密性定义基础上,可以把混合加密方案设计成具有可证明安全性的方案。在第 15 章,我们将评述几个这样的方案。

## 8.16 公钥密码系统密钥信道的建立

大家熟悉的 Diffie-Hellman 密钥交换协议(见 8.3.1 节)的中间人攻击在公钥系统中是很普通的。在一般情况下,为了发送一条用她/他的公钥加密的秘密消息给接收者,发送者必须首先确定要用到的这个密钥的确属于意定的接收者。同样,收到一个“数字信封”时,在使用从“信封”中得到的对称密钥进行秘密通信之前,接收者也必须确定这个“信封”确实是来自一个所声称的源。

因此,不管公钥密码技术有多么“不寻常”,仍然需要在通信双方之间建立一条安全的密钥信道。但是,在公钥密码学中,我们有  $ke \neq kd$ (见图 7.1),把加密密钥  $ke$  传送给消息的发送者不需要做任何秘密操作。所以,建立密钥信道的工作纯粹是一个认证问题,即密钥信道不包含任何的秘密操作,只需要保持加密密钥的认证性。

认证的密钥信道建立将是第 13 章的主要论题。基于公钥证书技术的公钥信道的建立将在 13.2 节介绍,而基于身份的技术将在 13.3 节介绍。

## 8.17 本章小结

在这一章,我们介绍了著名的并已得到广泛应用的公钥加密体制:Diffie-Hellman 密钥交换协议、RSA、Rabin 和 ElGamal 加密算法。在介绍这些基本的公钥加密体制的同时,我们也分别介绍了作为复杂性理论假设的困难问题,它们是基本公钥加密算法安全性的基础。

我们声明了这一章所考虑的安全性水准,为完全或无保密以及被动攻击者,是一种低水平的:以教科书式安全概念来标示,仅适合于数据总是随机的和不太聪明的攻击者(他们不实施主动攻击)的理想情况。本章所介绍的公钥体制全部是教科书式的。对它们所演示的各种攻击表明了它们的不安全性。

我们接着讨论了公钥加密体制需要更为严格并适合于应用的安全概念,以及对于在更强的安全定义下的安全体制的需求。但是,我们决定推迟到几个章节之后再做介绍(第 V 部分)。不打算学第 V 部分的读者,尤其是想运用本章介绍的教科书式密码体制的读者,应该仔细地复习本章给出的这些攻击。

## 习题

- 8.1 教科书式密码算法的两个显著特点是什么?
- 8.2 分组密码(在 7.8.2 节中介绍)的密码分组链接(CBC)模式有随机的输入,于是明文的任何部分信息可以被很好地隐藏起来。CBC 仍然是教科书式密码算法吗?为什么?
- 8.3 令对 Diffie-Hellman 密钥交换进行中间人攻击的攻击者仅在 Alice 和 Bob 之间转发消息(即“处于中间的人”除了用与 Alice 和 Bob 分享的密钥解密和加密外,不改变 Alice 和 Bob 的会话)。这种攻击是被动的还是主动的?  
提示:攻击在消息转发之前就发生了。
- 8.4 对于普遍认可的有限域  $\mathbb{F}_q$  规模设置的下界:  $|q| = 1204$ , 以及对于式(8.4.2)中的亚指数表达式  $\text{sub\_exp}(q)$  中的  $c < 2$ , 证明存在解决  $\mathbb{F}_q$  中 DLP 的“多项式时间算法”, 运行时间以  $q$  的 9 次多项式为界。
- 8.5 令  $\langle g \rangle$  有一个非保密的阶  $\text{ord}(g)$ 。下面的问题困难吗?  
给定  $g^c$ , 找  $g^a$  和  $g^b$  满足  $ab \equiv c \pmod{\text{ord}(g)}$ , 即由  $(g, g^c)$  构造一个 Diffie-Hellman 四元组  $(g, g^a, g^b, g^c)$ 。
- 8.6 离散对数问题与计算 Diffie-Hellman 问题有什么关系?
- 8.7 在 RSA 公钥数据  $(e, N)$  中, 为什么加密指数  $e$  必须与  $\phi(N)$  互素?
- 8.8 通常情况下分解奇合数是困难问题。那么分解素数的幂也是困难问题吗?(一个素数幂是  $N = p^i$ , 其中  $p$  是素数,  $i$  是整数。分解  $N$ )  
提示: 对任意  $i > 1$ , 计算  $N$  的  $i$  次根需要尝试多少个指数值  $i$ ?
- 8.9 假设  $N$  是一个素数幂, 前面问题中“计算  $N$  的  $i$  次根”的其中一种方法是二分查找。设计一个二分查找算法求  $p^i$  的  $i$  次( $i$  是知道的)根。证明这是一种有效的方法。  
提示: 考虑二分查找  $\frac{\log_2 N}{i}$  比特的素数。

- 8.10 RSA 加密函数是模 RSA 模数的乘群的一个置换,所以 RSA 函数也称为单向陷门置换。Rabin(ElGamal)加密函数是单向陷门置换吗?
- 8.11 令  $N \approx 2^{1024}$ 。在  $\mathbb{Z}_N^*$  中随机选取元素,选取的元素小于  $2^{64}$  的概率是多少? 用这个结果解释不应当把一个 64 比特的随机密码口令当做 RSA(Rabin、ElGamal)加密算法的随机明文的原因。
- 8.12 在什么情况下可以把 ElGamal 密码体制看做是确定的算法?
- 8.13 什么是 CPA、CCA、CCA2? 请解释这些概念。
- 8.14 我们用了“完全或无”作为对 RSA 和 Rabin 密码体制 CPA 安全性在描述上的一个改进(定理 8.1 和定理 8.2 (I))。为什么这是必须的?
- 8.15 为什么所有公钥加密算法(即使是教科书式加密算法)都必须抵抗 CPA?
- 8.16 教科书加密算法通常容易受到主动攻击,主要原因是什么?
- 8.17 什么是预言机(加密,解密)服务? 对于一个公钥加密算法,攻击者需要预言机提供加密服务吗?
- 8.18 由于教科书式密码算法通常容易受到主动攻击,我们已经建议应该注意不要提供任何(预言机)解密服务。这在实际上是正确的态度还是一个实用策略?
- 8.19 由于主动攻击通常要修改网络上传输的(密文)消息,那么如果公钥加密算法用了数据完整性检测技术来检测对密文消息的非授权修改,主动攻击会仍然有效吗?
- 8.20 混合密码体制的优点是什么?

## 第 9 章 理想情况下基本公钥密码函数的比特安全性

### 9.1 前言

在以前的章节中我们从几个实例已经看到,基本公钥密码函数一般不能很好地隐藏与明文相关的部分信息,尤其当一个明文不具有随机性时。然而,这些基本的密码原型函数本身的性质是相当好的,当它们在某种理想环境下(即明文消息是随机的情况下)是相当不错的。在这种情况下,它们中的每个基本函数实际上都是很强的。

在这一章里,我们将研究基本公钥密码函数的**比特安全性**。我们将看到在前面章节中所介绍的基本和通用的公钥密码原型函数都具有很强的比特安全性,这是因为,只要明文消息是随机的,那么从密文中恢复明文的单个比特就同恢复整个明文一样困难。

关于基本的和通用的公钥密码函数比特安全性的实际结果意味着,只要明文消息是随机的,那么恢复有关明文的任何消息的问题就同这些基本函数求逆一样困难,这是因为后者是恢复整个明文消息的问题。

许多研究者应用这个研究通过使用基本的和通用的公钥密码原型函数来构建很强的公钥加密方案。它的思想就是在应用一个原型函数之前,通过使用某种随机化方案把明文消息随机化。在第 V 部分,我们将研究称之为**随机预言机模型**的安全性证明的一般方法。在这个随机预言模型下,这些基于前一章所引入的通用公钥密码函数的公钥加密方案(事实上也包括了数字签名方案),在一个强安全概念下能够被证明是安全的。这类证明中的重要一招就是假设这些方案的输入明文已经被随机化了。

我们应该指出,“某种理想情况”就是明文消息是随机的情况。这样的情况并不是“理想情况”。后者,除了随机消息以外,Malice 也是一个从来不做主动攻击的“好人”。因此,基本的和通用的公钥密码原型函数在理想情况下仍然是很脆弱的。在这章我们将看到几个例子。

对于那些不想弄清楚将在第 V 部分介绍的、适合于应用的密码体制的读者,本章可以跳过。

#### 9.1.1 本章概述

9.2 节研究 RSA 的比特安全性。9.3 节研究 Rabin 比特安全性和利用 Rabin 比特生成一个强伪随机数的技术。9.4 节研究 ElGamal 的比特安全性。最后,在 9.5 节我们将对离散对数函数的比特安全性进行研究。

### 9.2 RSA 比特

如果一条 RSA 密文是对不包含事前可猜测的信息进行的加密(例如,当消息是 $\mathbb{Z}_N^*$ 上一个均匀分布的随机数)。那么众所周知,从密文中提取一比特的明文信息就同提取整个明文组一样困难[130,77,76]。不失一般性,我们考虑明文最低位比特,即明文消息的校验比特。在这里,我们将讨论下面的陈述:



**定理 9.1** 令  $N$  是一个 RSA 模数, 下面两个问题是同等困难的(或同等容易):

- I) 已知一个消息的 RSA 加密, 恢复消息。
- II) 已知一个消息的 RSA 加密, 恢复消息的最低比特。

如果能够求解问题(I), 那么显然也能够求解问题(II)。反之似乎没那么直接。可以认为这两个问题几乎不可能等价: 尽管明文消息是均匀随机分布的, 但是问题(I)是一个计算上的问题, 问题(II)是一个判定性问题, 并且单纯去猜测将会使人们有二分之一的机会。

然而, 如果一个人拥有能可靠回答问题(II)的预言机, 那么他通过对这个预言机的  $\log_2 N$  次询问, 的确能够求解问题(I), 我们将给出该方法。因为是  $N$  的二进制长度, 因此, 该方法在输入长度的多项式时间内能够把问题(I)归结为问题(II), 因而称为**多项式时间的归约**。因此, 若问题(I)能够以输入长度的多项式时间内解决, 同时就能知晓预言机求解问题(II)的时间。我们把这两个问题看做是具有相同的时间复杂度, 因为我们不能区别它们的复杂度的多项式表示有何不同之处。

现在, 我们来描述从问题(I)到问题(II)的多项式归约。我们称求解问题(II)的预言机为“RSA 奇偶预言机”, 并用  $PO_N$  表示, 即

$$m(\bmod 2) \leftarrow PO_N(m^e(\bmod N))$$

在定理 9.1 的证明中, 我们用  $x \in (a, b)$  表示在开区间  $(a, b)$  中一个整数, 其中  $a$  (或  $b$ ) 可能是整数也可能不是整数。因为是个整数,  $x \in (a, b)$  意味着  $x$  属于闭区间  $[a], [b]$ 。

证明的关键是一种二元搜索技术, 该技术可以由下面的引理得到。

**引理 9.1** 令  $N$  是一个奇数并且  $x \in (0, N)$ , 则  $2x(\bmod N)$  为偶数, 当且仅当

$$x(\bmod N) \in (0, \frac{N}{2})。$$

**证明** 对于所有  $x \in (0, \frac{N}{2})$ , 乘积  $2x(\bmod N)$  不用进行模运算。因此, 所得的结果是  $2x$  并且是  $(0, N)$  之间的一个偶数。相反, 如果  $2x(\bmod N)$  是一个偶数, 那么一定能够被 2 整除并且这个除法不用进行模运算。因此  $x \in (0, \frac{N}{2})$ 。□

因为所有的  $x \in (0, \frac{N}{2})$  占了区间  $(0, N)$  中的一半整数, 引理 9.1 也说明了  $2x(\bmod N)$  是一个奇数, 当且仅当  $x \in (0, \frac{N}{2})$ 。

现在让我们来证明定理 9.1。

**证明** 我们仅需证明问题(II)  $\Rightarrow$  (问题(I))。这个证明是构造性的。我们构造一个二元搜索算法。该算法利用一个可靠的  $PO_N$  并且能够从一个 RSA 密文  $c = m^e(\bmod N)$  求解出  $m$ 。该算法使用一个被称为“当前区间”的区间  $(a, b)$ , 简记为  $CI$ 。在算法的开始, 当前区间被设置为  $(a, b) = (0, N)$ 。这个元搜索算法保持下面两个不变条件:

- 在每次迭代中,  $CI$  区间减半
- 所解密的明文消息保留在  $CI$  区间中。

为了更清楚地说明, 我们将仅考虑这个搜索的前两次迭代过程。

**迭代 1** 我们知道明文是在区间  $(a, b) = (0, N)$  中。我们要求将  $2^e c \pmod{N}$  输入到  $PO_N$ 。注意,  $2^e c \equiv (2m)^e \pmod{N}$ 。由引理 9.1, 我们可以从  $PO_N(2^e c)$  推断出  $m \in (0, N - \frac{N}{2})$  还是  $m \in (0 + \frac{N}{2}, N)$ 。因此, 我们就获得了一个新的包含明文且长度减半的  $CI$ 。因此, 当进入这次迭代时,  $(a, b) = (0, N)$ ; 当我们完成这次迭代过程后, 我们就有  $(a, b) = (\frac{N}{2}, N)$  或者  $(a, b) = (0, \frac{N}{2})$ 。

**迭代 2** 我们考虑第一次迭代以后  $(a, b) = (\frac{N}{2}, N)$  的情况。假设我们将  $2^{2e} c \equiv (2^2 m)^e \pmod{N}$  送给  $PO_N$ 。如果  $PO_N(2^{2e} c) = 0$ , 那么明文  $2^2 m \equiv 4m \pmod{N}$  是偶数。由引理 9.1, 我们有  $2m \pmod{N} \in (0, \frac{N}{2})$ 。但是我们记得  $2m < 2N$ , 因此对  $2m \pmod{N} < \frac{N}{2}$  而言, 仅当  $2m < 2N - \frac{N}{2} = \frac{3N}{2}$  时, 也就是  $m < \frac{3N}{4}$  时, 它才成立。因此, 我们可以得到  $m \in (\frac{N}{2}, \frac{3N}{4})$  现在, 我们通过执行  $(a, b) \leftarrow (a, b - \frac{N}{2})$  来更新  $CI$ 。其余的迭代过程依此类推。

读者可以通过下面两种情况来检验  $CI$  是否被正确更新。

- 如果  $PO_N$  回答是 0, 那么, 明文是在区间  $CI = (a, b)$  的左半部分, 因此  $b$  被缩短了  $\frac{|CI|}{2}$ ;
- 否则, 明文是在区间  $CI = (a, b)$  的右半部分, 这时  $a$  被缩短了  $\frac{|CI|}{2}$ 。

很清楚, 当  $i = \lfloor \log_2 N \rfloor + 1$  次迭代以后, 运行达到  $|CI| = b - a < 1$  时, 这个搜索算法中止并输出明文  $m = b$ 。我们就结束了定理 9.1 的证明。□

算法 9.1 总结了在定理 9.1 的证明中构建的二元搜索算法的一般描述。

**例 9.1** 对于 RSA 公钥  $(N, e) = (15, 3)$ , 密文为  $c = 13$ , 假设我们向  $PO_N$  询问 4 个问题并指出秘密的明文  $m$ 。然后, 我们将下面看起来是随机密文的询问反馈给  $PO_N$ :

$$(2^3 \times 13, 4^3 \times 13, 8^3 \times 13, 16^3 \times 13) \equiv (14, 7, 11, 13) \pmod{15}$$

$PO_N$  回答: 0, 1, 1, 1。从这些回答中, 我们可以推断:

第一次回答 0  $\Rightarrow m \in (0, 15 - \frac{15}{2}) = (0, \frac{15}{2})$ , 也就是说,  $m \in [1, 7]$ ;

第二次回答 1  $\Rightarrow m \in (0 + \frac{15}{4}, \frac{15}{2}) = (\frac{15}{4}, \frac{15}{2})$ , 也就是说,  $m \in [4, 7]$ ;

第三次回答 1  $\Rightarrow m \in (\frac{15}{4} + \frac{15}{8}, \frac{15}{2}) = (\frac{45}{8}, \frac{15}{2})$ , 也就是说,  $m \in [6, 7]$ ;

第四次回答 1  $\Rightarrow m \in (\frac{45}{8} + \frac{15}{16}, \frac{15}{2}) = (\frac{105}{16}, \frac{15}{2})$ , 也就是说,  $m \in [7, 7]$ 。

因此, 我们已经求得  $m = 7$ 。事实上 7 的确是明文:  $7^3 = 13 \pmod{15}$ 。□

#### 算法 9.1 利用奇偶预言机进行二元搜索 RSA 明文

输入  $(N, e)$ : RSA 是公钥参数;

$c = m^e \pmod{N}$ : 是一 RSA 密文;

$PO_N$ : 一个奇偶预言机, 输入一 RSA 密文, 它返回相应于明文的最低位比特。

输出  $m$ 。



1. 初始化  $(a, b) \leftarrow (0, N)$ ; (\* “当前区间”的长度  $CI = (a, b)$  在每次迭代中取半, 但  $m \in (a, b)$  保持不变。\*)
2. For  $i = 1, 2, \dots, \lfloor \log_2 N \rfloor + 1$  do
  - {
  - (\*  $(a, b)$  的长度总是小于  $\frac{N}{2^{i-1}}$ 。\*)
  - 2.1 if  $(PO_N(2^i c) = 0)$  then  $b \leftarrow b - \frac{N}{2^i}$ ;
    - (\*  $m$  在  $(a, b)$  的左半部分 \*)
  - 2.2 Else  $a \leftarrow a + \frac{N}{2^i}$ ;
    - (\*  $m$  在  $(a, b)$  的右半部分 \*)
  - }
3. 返回  $(\lfloor b \rfloor)$ 。

定理 9.1 告诉我们 RSA 最低位比特的强度同整个明文一样。

我们从例 8.4 已经看到, 当一个 RSA 公钥的所有者充当一个解密预言机来返回一个明文作为解密请求的数据时, 这对于他而言是不安全的。现在由“RSA 最低位的安全性”的结果, 我们知道该用户也不能充当“奇偶预言机”或“ $N/2$  预言机”(由引理 9.1) 来回答相应于明文的奇偶比特的任何密码提问(或回答明文是否小于  $N/2$ )。

我们应该提醒读者, 攻击者可能把这样的提问嵌入到一个看似无害的协议中。请看下面的例子。

**例 9.2** 当 Alice 和 Malice 需要协商一个只有他们彼此共享的秘密会话密钥时, Malice 可能会提出一个似乎合理的建议:

“Alice, 我们彼此发送 1000 个用各自公钥加密的密文消息, 怎么样? 令会话密钥是由所交换的每对明文消息的校验比特进行异或运算所得的比特串。顺便说一下, 为了使你相信这个会话密钥是随机的, 先让我发送 1000 组密文给你。”

Alice 不仅同意, 她还感激 Malice 对她的信任(在生成随机密钥会话密钥中)! 然而, Malice 发给 Alice 的这 1000 个密文消息是  $(2^i)^e c \pmod N$ ,  $(i = 1, 2, \dots, 1000)$ , 其中  $c$  是其他人发送给 Alice 并且已被 Malice 窃听的密文。

这个协议完成后, Malice 谎称在会话密钥的计算过程中出错:

“Alice, 很对不起, 我的计算陷入困境, 麻烦你把会话密钥发送给我, 行吗? 请用我的公钥加密”。

可怜的 Alice 提供了帮助。Malice 能够由会话密钥提取所需要的比特, 然后应用算法 9.1 就能提取  $c$  中被加密的明文。□

在这里 Malice 是一个主动的攻击者: 他能够通过乘以因子  $(2^i)^e \pmod N$  修改密文。因此, 尽管 RSA 的最低比特的强度同整个明文消息一样, 这个函数仍然无力抵抗主动攻击。

## 9.3 Rabin 比特

为了使算法 9.1 能够应用于 Rabin 加密中,在加密时,我们可以将  $c$  修改为  $c = m^2 \pmod{N}$  形式即可(也就是说,我们仅仅修改指数,即加密指数是  $e = 2$  的情况)。

然而,情况有点复杂。由于  $N$  有两个不同的素因子,由定理 6.17(见 6.6.2 节)可知,任意  $c \in QR_N$  有四个不同的模  $N$  平方根,即密文有四个不同的明文。如果一个奇偶预言机只能回答  $c$  的一个随机的平方根,那么,这个预言机就不可靠了,因而,也就不能使用了。然而,如果一个平方根有某个特殊的性质,能够使预言机确定性地工作(具有可靠性),那么,这个二元搜索方法仍能用于 Rabin 加密。

一个回答 Jacobi 符号为正的最小平方根的预言机是一个具有确定性的预言机实例。由定理 6.18(见 6.7 节)我们知道,如果  $N$  是一个 Blum 整数,那么任意一个二次剩余  $c$  有两个 Jacobi 符号为正的根:  $m, -m$ 。因为  $N$  是奇数,这两个根中只能有一个小于  $\frac{N}{2}$ ,因此我们称这个根为“ $c$  的 Jacobi 符号为正的较小根”。

现在,如果我们把  $N$  限制为是一个 Blum 整数,那么有  $\left(\frac{2}{N}\right) = 1$  ( $N = pq$  且  $p \equiv q \pmod{N}$ ),因而一个输入 Jacobi 符号为正的最小平方根的奇偶预言机就可以工作了。注意,  $\left(\frac{2}{N}\right) = 1$ 。对这个可靠的奇偶预言机的第  $i$  次提问所得到的明文是  $\frac{m}{2^i} \pmod{N}$ ,因此要保存所有  $i$  次提问明文的 Jacobi 符号的符号。对为使 Rabin 加密而修订的二元搜索算法有兴趣的读者,请参考[130]。

### 9.3.1 Blum-Blum-Shub 伪随机比特生成器

能使 Rabin 加密的二元搜索算法这一事实表明:如果 IF 假设(8.8 节中的算法 8.4)成立,那么 Rabin 的最低比特位是很强的。Rabin 的最低比特位的强度有一个重要的应用:密码强伪随机比特(CSPRB)的生成[43]。这个所谓的 **BBS 伪随机数生成器** 用一个种子  $x_0 \in QR_N$ , 其中  $N$  是一个  $k$  比特的 Blum 整数。那么,这个用  $x_0$  做种子的 BBS 生成器生成的伪随机比特是由下面这个序列中的每个数的最低比特组成:

$$x_0, x_1 = x_0^2, \dots, x_i = x_{i-1}^2, \dots \pmod{N} \quad (9.3.1)$$

可以证明[43,130],不知道种子  $x_0$ , 预言式(9.3.1)中序列的最低比特在计算上等价于分解 Blum 整数  $N$ 。

**注释 9.1** 众所周知[13,297],从一条 Rabin 加密密文中同时提取  $\log_2 \log_2 N$  个最低比特位等价于分解大整数  $N$ 。□

Blum 和 Goldwasser 应用这个结果并提出了一个有效的密码体制,该密码体制具有很强的安全性,称之为语义安全性。我们将在 14 章中介绍语义安全性,同时在那里我们还将介绍基于 Rabin 比特强壮性的 Blum 和 Goldwasser 语义安全的密码体制。

## 9.4 ElGamal 比特

对于算法 8.3 中所提出的 ElGamal 密码体制而言,因为明文消息空间是  $\mathbb{F}_p^*$ , 其中  $p$  是一个

大素数(也是奇数),因此,二元搜索技术也能够被直接应用。为了能够从密文对 $(c_1, c_2)$ 找到对应的明文消息,那么输入到奇偶预言机的提问密文消息应该是:

$$(c_1, 2^i c_2) \pmod{p}, \text{ 其中 } i = 1, 2, \dots, \lfloor \log_2 p \rfloor + 1$$

如果奇偶预言机是一个人(这是很可能的,如例 9.2),那么为了避免被怀疑,攻击者可以盲化这些提问,例如:

$$(g^{r_i} c_1, 2^i y^{r_i} c_2) \pmod{p} \text{ 其中 } r_i \in \mathbb{Z}_{p-1}, i = 1, 2, \dots, \lfloor \log_2 p \rfloor + 1$$

其中 $(g, y)$ 是奇偶预言机的公钥部分。这些 $\lfloor \log_2 p \rfloor + 1$ 个密文消息对是完全彼此独立的。然而,它们加密了相关消息列 $2^i m \pmod{p}$ ,其中 $i = 1, 2, \dots, \lfloor \log_2 p \rfloor + 1$ :

$$(g^{k+r_i}, y^{k+r_i} 2^i m) \pmod{p}, i = 1, 2, \dots, \lfloor \log_2 p \rfloor + 1$$

最后,我们可推断出 ElGamal 密码体制的比特安全性同整个数据块的安全性一样困难。另一方面,一个公钥的持有者应该注意,别像例 9.2 游戏中的 Alice 那样被欺骗。

## 9.5 离散对数比特

在 8.4 节中,我们已经讨论了在一般情况下,阿贝尔群的离散对数问题是困难的:相信函数 $g^x$ 具有单向性。此外,目前还不清楚这个函数是否是一个陷门函数。因此,在预言机的帮助下,从 $g^x$ 提取 $x$ 是个奇怪的想法。然而,为了研究离散对数函数的比特安全性与整组安全性的关系,我们假设存在一个预言机,当输入一个 $(g, g^x)$ 对时,能够回答的一些比特级的部分信息。

假设元素的阶为奇数,如果 $\text{ord}(g)$ 是已知的(这是通常情况),那么 $\frac{1}{2} \pmod{\text{ord}(g)}$ 就是可求的。在这种情况下,通过奇偶预言机逐比特来提取离散对数问题事实上就成了模指数运算(见算法 4.3)的逆运算。因为模指数算法也称为“平方-乘法”方法,逆运算应该称为“开方-除法”方法。算法 9.2 将详细地说明该方法。

如果元素 $g$ 的阶为偶数,那么将会怎么样?例如, $g$ 是 $\mathbb{F}_p^*$ 的一个生成元,其中 $p$ 为素数,那么有 $\text{ord}_p(g) = p - 1$ 是偶数且不与 2 互素。因此, $\frac{1}{2} \pmod{p-1}$ 不存在。因此,我们不能用算法 9.2 中的步骤 2.2 中的方式来求解 $h$ 的平方根。

### 算法 9.2 利用奇偶预言机求解离散对数

输入  $(g, h)$ :  $g$  是阶为奇数的群元素,且满足  $h = g^x$ ;

$PO_{\text{desc}(g)}$ : 一个奇偶预言机,  $PO_{\text{desc}(g)}(g, h) = \log_g h \pmod{2}$ 。

输出 整数  $x$ 。

1. 假设  $x \leftarrow 0; y \leftarrow \frac{1}{2} \pmod{\text{ord}(g)}$ ;

2. 重复下面的步骤直到  $h = 1$  (\* 包括  $h = 1$  \*)

{

2.1 if(  $PO_{\text{desc}(g)}(g, h) = 1$  ) then  $h \leftarrow h/g; x \leftarrow x + 1$ ;

(\* 当  $\log_g h$  是奇数时,在模指数下,“除并加 1”是“乘并减 1”的逆 \*)

2.2  $h \leftarrow h^y; x \leftarrow 2x;$

(\* 当  $\log_g h$  是偶数时, 在模指数下, “取平方根并加倍”是“开平方并减半”的逆 \*)

}

3. 返回( $x$ )。

然而, 在这种情况下( $g$  是  $\mathbb{F}_p^*$  中的一个生成元), 我们仍然可以运用算法 6.4 来计算  $h$  模  $p$  的平方根。对于任意的二次剩余元  $h \in \text{QR}_p$ , 平方根算法将返回两个  $h$  的平方根, 我们将它们表示为  $\pm\sqrt{h}$ 。因  $g$  为是  $\mathbb{F}_p^*$  的一个生成元,  $g \in \text{QR}_p$  成立; 而  $h \in \text{QR}_p$ , 因此  $\log_g h$  一定是一个偶数。不失一般性, 我们可以把以  $g$  为底的  $h$  的两个平方根的离散对数写成下面的形式:

$$\log_g \sqrt{h} = \frac{\log_g h}{2}, \log_g (-\sqrt{h}) = \frac{p-1}{2} + \frac{\log_g h}{2} \quad (9.5.1)$$

注意, 因为式(9.5.1)中的加法是模  $p-1$  运算的, 在式(9.5.1)的两个数中恰有一个小于  $\frac{p-1}{2}$ , 另一个一定大于等于  $\frac{p-1}{2}$ 。显然, 以  $g$  为底有较小离散对数的平方根就是正确的平方根。

麻烦的是, 我们不能从  $\sqrt{h}$  和  $-\sqrt{h}$  看出哪个平方根有以  $g$  为底的更小离散对数!

然而, 如果我们有一个不同的“1 比特信息的预言机”, 当输入( $g, y, -y$ )时, 该预言机可以输出  $y$  或  $-y$ , 不管哪一个有以  $g$  为底的最小离散对数, 我们都可以用这个预言机(称它为半阶预言机)来为我们选择正确的平方根。算法 9.3 就是通过修正算法 9.2 做这个工作的。

### 算法 9.3 利用半阶预言机求解离散对数

输入 ( $g, h, p$ ):  $g$  是  $\mathbb{F}_p^*$  的生成元, 其中  $p$  为素数;  $h = g^x \pmod{p}$ ;

$PO_{(p,g)}$ : 一个半阶预言机;

$$PO_{(p,g)}(g, y, -y) = \begin{cases} y & \text{若 } \log_g y < \log_g (-y) \\ -y & \text{其他} \end{cases}$$

输出 整数  $x$ 。

1.  $x \leftarrow 0$ ;

2. 重复下面的步骤直到  $h = 1$  (\* 包括  $h = 1$  \*)

{

2.1 if( $h \in \text{ONR}_p$ ) then  $h \leftarrow h/g; x \leftarrow x + 1$ ;

(\*  $h \in \text{ONR}_p$  意味着  $\log_g h$  是奇数; 这一点可以通过验证 Legendre 符号  $(\frac{h}{p}) = -1$  来完成。\*)

2.2  $h \leftarrow PO_{(p,g)}(g, \sqrt{h}, -\sqrt{h}); x \leftarrow 2x$ ;

(\* 我们可以不费力地进行平方根运算, 但是需要预言机来告诉我们哪个根是正确的, 即它有较小的离散对数  $\frac{\log_g h}{2}$ 。\*)

}

3. 返回( $x$ )。

由于检测 Legendre 符号和计算平方根模一个素数可以有效地进行,从算法 9.3 我们可知,从  $(g, h)$  中决定  $\log_g h$  是否小于  $\frac{\text{ord}(g)}{2}$  等价于根据  $(g, h)$  来求解  $\log_g h$ 。

算法 9.3 比算法 9.2 更一般化,因为阶数为奇数  $g$  的也可在算法 9.3 上使用。下面让我们看一个例子。

**例 9.3** 假设我们有一个“半阶预言机”。群  $\mathbb{Z}_{23}^*$  的生成元  $g = 5$  和元素  $h = 9$ 。我们通过“半阶预言机”来求解。

当输入为  $(5, 9, 23)$  时,算法 9.3 的运行如下。双箭头表示“开平方根”,其中水平方向的双箭头( $\Rightarrow$ )是由“半阶预言机”所做的选择。单箭头( $\rightarrow$ )代表“除以  $g$ ”( $g = 5$ )。所有的计算都是在模 23 上运行的。

$$\begin{array}{ccccccccccc} 9 & \Rightarrow & 20 & \rightarrow & 4 & \Rightarrow & 2 & \Rightarrow & 5 & \rightarrow & 1 & \Rightarrow & 1 \\ \Downarrow & & & & \Downarrow & & \Downarrow & & & & \Downarrow & & \\ 3 & & & & 21 & & 18 & & & & 22 & & \end{array}$$

在算法的开始,  $x$  被初始化为 0 (在第 1 步); 对每个双箭头  $\Rightarrow$ , 表示在第 2.2 步中进行  $x \leftarrow 2x$  操作; 对每个单箭头  $\rightarrow$ ; 表示在第 2.1 步中进行  $x \leftarrow x + 1$  操作。当算法中止时,最后输出  $x$  的值为 10。事实上  $9 = 5^{10} \pmod{23}$ 。□

这些结果表明,通常情况下离散对数的单个比特同整个组具有同样的安全性。我们也知道,如果一个生成元是二次剩余的话,那么以这个元素为底的离散对数的所有比特(包括最低位比特)都是困难的。这就引出了 ElGamal 密码体制的一种语义安全的版本。这种密码体制将在 14 章中介绍。

## 9.6 本章小结

对于对基本和通用的公钥密码算法比特级的安全性强度的研究,在不断地取得许多积极的成果:在这些函数的作用下,恢复明文中每比特的困难性等同于恢复整个明文组的困难性。这些积极的成果提出了下面的观点:如果一个明文消息是随机的,那么找有关明文的任何信息同求这些基本函数的逆一样困难。

许多研究者利用这一结果构建了比基本和通用的公钥密码原型更强的公钥加密方案。其思想是利用一些随机化方案来随机化明文消息。在第 V 部分,基于基本的和通用的公钥密码原型,我们将研究一个通用的方法,即用随机预言模型来完成强的和可证明安全的公钥加密方案(事实上,也含数字签名方案)的构造。

通过对几种基本和通用的公钥密码函数的研究,我们看到了这些函数的一个一成不变的弊端:它们对于主动攻击极为脆弱。在第 V 部分将要研究强化公钥加密算法的一般方法,也包括挫败主动攻击的一些机制。

## 习题

9.1 完成定理 9.1 的证明中“迭代 2”的其余 3 种情况。

i)  $(a, b) = \left(\frac{N}{2}, N\right), PO_N(2^{2^e}c) = 1;$

ii)  $(a, b) = \left(0, \frac{N}{2}\right), PO_N(2^{2^e}c) = 0;$

iii)  $(a, b) = \left(0, \frac{N}{2}\right), PO_N(2^{2^e}c) = 1。$

9.2 在什么条件下 RSA 的加密算法具有很强的比特安全性?

提示:如果一个明文有某些可验证的部分信息,加密算法具有很强的整组安全性吗?

9.3 基本的公钥加密算法的强比特安全性意味着这些算法是安全的吗?

9.4 BBS 伪随机数生成器的安全基础是什么?

9.5 令  $p$  是个大素数,  $g$  是  $\mathbb{F}_p^*$  的一个生成元,容易计算  $g^x \pmod{p}$  的 Legendre 符号意味着容易计算  $x$  的奇偶比特。从  $g^x \pmod{p}$  中提取  $x$  为什么仍然是一个困难问题呢?



## 第 10 章 数据完整性技术

### 10.1 引言

在第 2 章,我们关于开放通信网络的脆弱性做了一个理想的、标准的假设:所有的通信都经受一个称为 Malice 的攻击者,他可以随意地窃听、截取、重发、修改、伪造或插入消息。当 Malice 插入了修改过的或伪造的消息,他将试图欺骗目标接收者,使其相信该消息来自某个其他合法的主体。安全电子商务交易,或以消息认证形式提供安全服务的密码机制(抵抗被动窃听)等中所要求的,以安全的方式使用这种脆弱的通信媒介是不够的。我们需要一种机制,使得消息的接收者可以验证该消息确实是来自所声称的消息源,且在传输的过程中未受到未授权方式修改。**数据完整性**就是抗击对消息未授权修改的安全服务。

现代密码学中的数据完整性与通信学中的一个经典的主题检错码有密切的联系,并由其演变而来。检错码是检测由于通信的缺陷而导致消息发生错误的方法。通常认为,使用被恶意方式修改过的信息和使用由于通信或数据处理的不当而导致的错误信息是同样危险的。因此,数据完整性技术的工作原理和检错码技术的工作原理本质上是相同的:消息的发送者通过编码为消息增加一些冗余来生成一个“校验值”,并将该校验值附在消息之后;消息的接收者根据与发送者协商好的一系列规则,利用附加的校验值来检验所接收到的消息的正确性[277]。在检错码中,通过这种编码加入的冗余度使得消息的接收者可以采用极大似然检测器来判定接收到的码字应该译为哪条消息,这条消息就是最可能由或许被改动过的码字所发送的消息。在数据完整性保护中,以这种方式编码加入的冗余使得加入的校验值在整个校验值空间中尽可能地均匀分布,这就使得攻击者伪造一个有效校验值的概率达到最小。对后一种增加冗余度的密码变换的方式类似于我们在 7.1 节描述的加密的混合变换性质,虽然对加密来说,混合变换不是基于增加可验证的冗余度。

与加密算法类似,实现数据完整性的密码转换也应该由密钥参数控制。因此,在通常意义下,一个正确的数据完整性验证结果也将向验证者提供有关消息源,亦即生成该数据完整性保护的主体的知识。但是,最近出现了一个新的概念即“无需源识别的数据完整性”。这个新的概念对于研究抗击适应性攻击的公钥密码系统是很重要的。我们将用一个例子来介绍这个概念,这个例子将为后面章节研究抵抗适应性攻击的公钥密码系统做准备。

#### 10.1.1 本章概述

首先,我们通过给出数据完整性保护的一个语法定义(10.2 节)来开始本章的技术内容部分。将对提供数据完整性服务的密码技术做一介绍。介绍分为对称技术(10.3 节)、非对称技术(10.4 节)和无识别源的数据完整性概念(10.5 节)。

### 10.2 定义

**定义 10.1 数据完整性保护** 设 Data 为任意信息,  $K_e$  为编码密钥,  $K_d$  为与该编码密钥相匹配

的验证密钥。Data 的数据完整性保护包括下面的密码变换：

篡改检测码的生成：

$$\text{MDC} \leftarrow f(\text{Ke}, \text{Data})$$

篡改检测码的验证：

$$g(\text{Kv}, \text{Data}, \text{MDC}) = \begin{cases} \text{True, 概率为 1} & \text{如果 } \text{MDC} = f(\text{Ke}, \text{Data}) \\ \text{False, 压倒性概率} & \text{如果 } \text{MDC} \neq f(\text{Ke}, \text{Data}) \end{cases}$$

其中  $f$  和  $g$  都是有效的密码变换：前者由一个辅助输入  $\text{Ke}$  (编码密钥) 参数化, 后者由任意输入  $\text{Kv}$  (验证密钥) 参数化; MDC 是 Manipulation Detection Code (篡改检测码) 的缩写。如果签字/验证算法是概率算法, 那么概率空间<sup>①</sup> 包括 Data、MDC 和密钥的所有可能情况, 可能还包含一个随机输入空间。

图 10.1 给出了数据完整系统的一个示例。

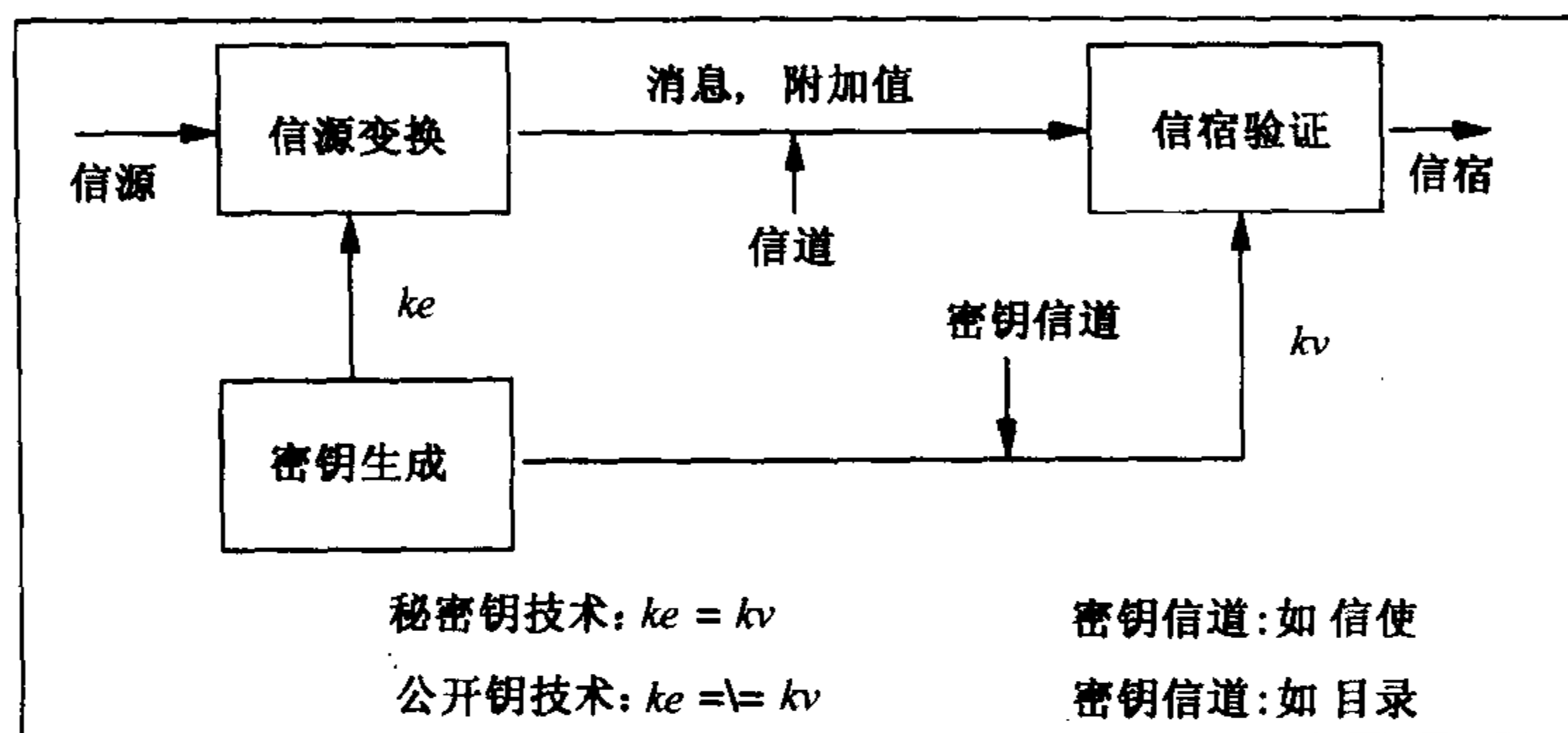


图 10.1 数据完整性系统

我们应该注意到, 尽管在我们的介绍性讨论 (和图 10.1) 中, 采用了通信场合来引入数据完整性保护概念, 然而定义 10.1 并不局限于通信场合; 例如, 数值对 (Data, MDC) 可以被存储到不安全的存储器中, 或从一个不安全的数据存储器中恢复。

类似于密码系统的情况, 数据完整性保护技术也可以分为对称技术和非对称技术。然而对于公钥技术, 我们应当注意区分这两种系统的差异。在由非对称技术实现的密码系统中, 公钥和私钥有固定的用法: 公钥用于消息编码 (加密), 私钥用于消息解码 (解密)。而在由非对称技术实现的数据完整性系统中, 公钥 (私钥) 既可以用于编码, 又可以用于验证。这两种不同的用法将分别是 10.4 节和 10.5 节讨论的主题。

### 10.3 对称技术

在实现数据完整性的对称技术中, 密码变换  $f$  和  $g$  (见定义 10.1) 是对称密码算法, 这意味着  $f = g$ , 并且  $\text{Ke} = \text{Kv}$ 。也就是说, Data 和 DMC 之间一致性的生成和验证采用相同的密码操作。

<sup>①</sup> 这里“压倒性”概率的含意由 4.6 节中所定义的“压倒性”概念而来。

由于数据完整性和消息认证(我们将在第 11 章研究消息认证)之间的密切关系,由对称密码技术生成的 DMC 常称为消息认证码(MAC)。MAC 的生成和验证可以使用密钥杂凑函数技术,也可以使用分组密码加密算法。

### 10.3.1 密码杂凑函数

实现 MAC 通常的方法是使用所谓的密钥杂凑函数技术。我们首先介绍密码杂凑函数。

杂凑函数是一个确定的函数,它将任意长的比特串映射为定长比特串的杂凑值。设  $h$  表示一个杂凑函数,其固定的输出长度用  $|h|$  表示。我们希望  $h$  具有以下性质:

#### 性质 10.1 杂凑函数的性质

- **混合变换** 对于任意的输入  $x$ ,输出的杂凑值  $h(x)$  应当和区间  $[0, 2^{|h|}]$  中均匀的二进制串在计算上是不可区分的。这里,计算上的不可区分性是由定义 4.15(见 4.7 节)而来。由假设 4.2(也见 4.7 节)可知,该性质是合理的。
- **抗碰撞攻击** 找两个输入  $x$  和  $y$ ,且  $x \neq y$ ,使得  $h(x) = h(y)$ ,在计算上应当是不可行的。为使这个假设成立,要求  $h$  的输出空间应当足够的大。 $|h|$  最小为 128,而典型的值为 160。
- **抗原像攻击** 已知一个杂凑值  $h$ ,找一个输入串  $x$ ,使得  $h = h(x)$ ,在计算上是不可行的。这个假设同样也要求  $h$  的输出空间足够大。
- **实用有效性** 给定一个输入串  $x$ ,  $h(x)$  的计算可以在关于  $x$  的长度规模的低阶多项式(理想情况是线性的)时间内完成。

杂凑函数的混合变换和抗碰撞性可以使用与分组密码算法设计(见 7.6 节 ~ 7.7 节)中所用的类似操作来实现。抗原像攻击可以通过一些数据压缩技术实现,数据压缩技术损失一部分输入数据,因而使得该函数不可逆。

我们不准备描述任何具体杂凑函数的设计细节。好学的读者可以在相关的文献中找到这些内容(如[200]的第 9 章)。

#### 10.3.1.1 杂凑函数在密码学中的应用

杂凑函数广泛应用于密码学中。这里我们列出杂凑函数的几个重要用途:

- 在数字签名中,杂凑函数一般用来产生“消息摘要”或“消息指纹”。这种用法是为将要签署的消息增加一个可以验证的冗余,以便这个杂凑消息包含可以识别的信息。我们将在本章(10.4 节)的数字签名中看到杂凑函数的这种一般用法。在那里,我们将主要依赖包含在签名消息中的一些可识别的冗余信息来实现数字签名体制的安全性(不可伪造性)。在第 16 章,我们将给出形式化证明,证明杂凑函数的这种用法为数字签名体制提供可证明安全性。
- 在具有实用安全性的公钥密码系统中,杂凑函数被广泛地用于实现密文正确性验证机制。对于要获得可证明安全的抗主动攻击的加密体制来说,这个机制是必不可少的。在本章(10.5 节),我们将看到关于这种用法的一个例子。在第 15 章,我们将给出形式化的证明,证明杂凑函数的这种用法能够为公钥加密提供可证明安全。在那里,我们还会进一步看到,为使公钥加密是可证明安全的,杂凑函数所起的一些更基本的作用。

- 在需要随机数的密码学应用中,杂凑函数被广泛地用做实用的伪随机函数。这些应用包括:密钥协商(如两个主体将他们自己的随机种子作为杂凑函数的输入,得到一个共享的密钥值),认证协议(如协议双方通过交换某些杂凑值来证实协议执行的完整性),电子商务协议(如以赌博方式实现小额支付的聚集),知识证明协议(如实现非交互式的证明,见 18.3.2.2 节)。在本书的其他部分,我们将看到杂凑函数这些用法的大量例子。

### 10.3.1.2 随机预言机

让我们扼要重述杂凑函数的“混合变换”性质:对于任意的输入,输出杂凑值的分布和函数输出空间上的均匀分布在计算上是不可区分的。如果我们将“与均匀分布在计算上不可区分”变成“是均匀的”,那么我们就将该杂凑函数变成一种很强的、虚构的函数,称之为**随机预言机**。

我们将随机预言机看成一个很强的函数,是因为它组合了以下这三种性质,即确定性、有效性和均匀输出。我们之所以将随机预言机称为虚构的函数,是因为就我们所知道的计算模型中,还没有如此强大的计算机制或机器。

一方面,我们知道如何有效地输出均匀分布的随机值,如抛掷一个公平的硬币。但是,这种输出随机数的方法不是一个确定的过程。另一方面,我们也可以为一组均匀独立的值赋予某种确定的关系,如在分类表中它们之间的距离。但是,这种关系不能在随机值规模的多项式时间内计算出(查找  $n$  项需要  $n \log n$  步)。

实际上,随机预言机的确定性和均匀输出特性意味着随机预言机输出的熵大于其输入的熵(回忆 3.7 节中熵的定义)。但是,根据香农的熵的理论(见 3.7 节中定理 3.2),一个确定函数决不可能“放大”熵。因此,在真实的环境中不存在随机预言机。

因为杂凑函数的混合变换性质仅是一个计算上的假设(4.7 节中的假设 4.2),因此在真实的环境中,杂凑函数应当具有这一性质,它仅等价于由定义 4.15(见 4.7 节)所给出的计算上不可区分性。也就是说,它的输出满足某种概率分布(在输出消息空间中),使得可以不被多项式限定的分辨者区分。因此,真实环境中的杂凑函数仅仅以某种精度仿真随机预言机的行为,希望它们之间的差异是一个可以忽略的量。

然而,杂凑函数仿真随机预言机的行为在公钥密码系统中扮演着重要的角色。从本质上说,对一个消息求杂凑值就是以确定的可验证的方式为该消息增加一定的冗余量。

### 10.3.1.3 生日攻击

假设一个杂凑函数  $h$ ,其真正的行为如同一个随机预言机,平方根攻击(生日攻击,见 3.6 节)说明,杂凑函数的  $2^{l_h/2} = \sqrt{2^{l_h}}$  个随机杂凑值足以使攻击者可以一个不可忽略的概率得到一个碰撞。为了实施生日攻击,攻击者应当生成消息-杂凑值对

$$(m_1, h(m_1)), (m_2, h(m_2)), \dots$$

直到找到两个消息  $m$  和  $m'$ , 满足

$$m \neq m', h(m) = h(m') \quad (10.3.1)$$

这样的一对消息称为杂凑函数  $h$  的碰撞。当然,为了使生日攻击对攻击者有用,碰撞消息  $m$  和  $m'$  应当包含一些有意义的子消息。例如,令一个要进行杂凑的消息(和数字签名,见 10.4 节)是下面形式的支付授权语句

$$M = \text{Price}, \text{Goods\_Description}, R$$

其中,  $R$  是一个随机数, 使得该协议消息随机化(总是希望协议消息是随机化的)。那么, 一个有趣的生日攻击可以是

$$m = \text{Price\_1}, \text{Goods\_Description}, r$$

和

$$m' = \text{Price\_2}, \text{Goods\_Description}, r'$$

其中,  $\text{Price\_1} \neq \text{Price\_2}$  和  $\text{Goods\_Description}$  是固定的消息部分, 碰撞是指关于随机数  $r \neq r'$  的碰撞。我们可以将

$$h'(x) \stackrel{\text{def}}{=} h(\text{Price\_1}, \text{Goods\_Description}, x)$$

和

$$h''(x) \stackrel{\text{def}}{=} h(\text{Price\_2}, \text{Goods\_Description}, x)$$

看成两个随机函数。因此在这样的消息中寻找一个碰撞和在式(10.3.1)的随机消息中寻找一个碰撞具有相同的复杂度。

很明显, 如果杂凑函数不是真正的随机函数, 所需计算的函数值就会更少。

因此, 密码杂凑函数的输出空间大小必须有一个下界。目前, 在应用密码学中广泛使用的杂凑函数是 SHA-1[219] 和 RIPEMD-160[54], 它们的输出长度都是  $|h| = 160$ , 抗平方根攻击的强度也都为  $2^{80}$ 。这与密钥长 80 比特的分组密码算法的强度相一致。以前通用的杂凑函数 MD5[245] 输出长度为  $|h| = 128$ , 调整后适合 DES 的 56 比特密钥长度和 64 比特的分组长度。

在介绍 AES-128、AES-192 和 AES-256 的同时(密钥长度分别为 128 比特、192 比特和 256 比特的 AES, 见 7.7 节), 标准化组织(如 ISO/IEC[153])正在对杂凑函数制定标准, 使之符合输出长度  $|h| \in \{256, 384, 512\}$ 。

### 10.3.2 基于密钥杂凑函数的 MAC

密码杂凑函数自然地成为数据完整性的一种密码原型。在共享密钥情况下, 杂凑函数将密钥作为它的一部分输入, 另一部分输入为需要认证的消息。因此, 为了认证一个消息  $M$ , 发送者计算

$$\text{MAC} = h(k \parallel M)$$

其中,  $k$  为发送者和接收者的共享密钥, “ $\parallel$ ”表示比特串的连接。

依据 10.3.1 节所列的杂凑函数的性质, 我们可以假设, 为了用杂凑函数关于密钥  $k$  和消息  $M$  生成一个有效的 MAC, 该主体必须拥有正确的密钥和正确的消息。与发送者共享密钥  $k$  的接收者应当由所接收的消息  $M$  重新计算出 MAC, 并检验同所收的 MAC 是否一致。如果一致, 就可以相信该消息来自所声称的发送者。

因为这样的 MAC 是使用杂凑函数构造的, 因此也称为 HMAC(用杂凑函数构造的 MAC)。为谨慎起见, HMAC 通常通过下面的形式计算:

$$\text{HMAC} = h(k \parallel M \parallel k)$$

也就是说, 密钥是要认证消息的前缀和后缀[290], 这是为了阻止攻击者利用某些杂凑函数的“轮函数迭代”结构。如果不用密钥保护消息的两端, 某些杂凑函数所具有的这种已知结构, 使得攻击者不必知道密钥  $k$  就可以选择一些数据用做消息前缀或后缀来修改消息。



### 10.3.3 基于分组加密算法的 MAC

构造密钥杂凑函数的标准方法是使用分组密码算法的 CBC 运行模式。通常,这样构造的密钥杂凑函数称为 MAC。

令  $\mathcal{E}_k(m)$  表示输入消息为  $m$ 、密钥为  $k$  的分组密码加密算法。为了认证消息  $M$ , 发送者首先对  $M$  进行分组:

$$M = m_1 m_2 \cdots m_\ell$$

其中,每一个子消息组  $m_i (i = 1, 2, \cdots, \ell)$  的长度都等于分组加密算法输入的长度。如果最后一个子消息组  $m_\ell$  长度小于分组长度,就必须对其填充一些随机值。设  $C_0 = IV$  为随机初始向量。现在,发送者用 CBC 加密:

$$C_i \leftarrow \mathcal{E}_k(m_i \oplus C_{i-1}), i = 1, 2, \cdots, \ell$$

然后,数值对

$$(IV, C_\ell)$$

作为 MAC 将附在  $M$  后送出。

很明显,在生成 CBC-MAC(由运行 CBC 模式的分组密码构造的 MAC)的计算中包括了不可求逆的数据压缩(本质上, CBC-MAC 是整个消息的“短摘要”),因此 CBC-MAC 是一个单向变换。而且,所用的分组密码加密算法的混合变换性质为这个单向变换增加了一个杂凑特点(也就是说,将 MAC 分布到 MAC 空间与分组密码加密算法应该将密文分布到密文空间同样均匀)。因此,我们可以设想为了生成一个有效的 CBC-MAC,该主体必须知道控制分组密码算法的密钥  $k$ 。与发送者共享密钥  $k$  的接收者应当由所接收的消息  $M$  重新计算出 MAC,并检验与所收的 MAC 是否一致。如果一致,就可以相信该消息来自所声称的发送者。

我们有时用  $\text{MAC}(k, M)$  表示一个 MAC,它为共享密钥  $k$  的主体的消息  $M$  提供完整性服务。在这个表示法中,我们忽略了实现细节,比如,为实现 MAC 采用了何种单向变换。

## 10.4 非对称技术I:数字签名

在公钥密码体制中,一个主体可以使用她/他自己的私钥“加密”消息,所得到的“密文”可以用该主体的公钥“解密”来恢复成原来的消息。很明显,如此生成的“密文”和“被加密”的消息一起可以起到篡改检测码(MDC)的作用,即为消息提供数据完整性保护。这里的公钥“解密”过程是 MDC 验证过程的一个步骤。

而且,因为任何人都可以得到公钥,也就都可以执行这样一个 MDC 的验证过程,然而,只有用于 MDC 验证的那个公钥的主人才能用相应的私钥生成 MDC。因此,公钥密码的这种用法可以精确地对证明是消息原作者的签名即数字签名的性质建立模型。换句话说,公钥密码更精确地讲是一个单向陷门函数(见 8.1 节的性质 8.1),可以用来实现数字签名机制<sup>①</sup>。Diffie 和 Hellman 首先提出了数字签名概念[98](这篇论文于 1976 年发表,但第一次是在 1975 年以预印的形式发表,见[97])。

<sup>①</sup> 尽管对数字签名来说,更为基础的是单向函数,见[175],但是对实用的数字签名来说,其基础是单向陷门函数。

能够提供数字签名是公钥密码相对于私钥密码的一个很大的优点(公钥密码的另一个重要的优点是能够实现远程主体之间的密钥分配,见 8.15 节)。既然只有一个实体可以生成消息的一个数字签名,并可以被任何人所验证,所以很容易处理关于是谁生成了该签名的纠纷。这一点可以提供一种安全服务,称为不可否认性,意味着不可否认与一个消息的关系。在电子商务应用中,不可否认性是必不可少安全要求。

定义 10.2 从语法形式上给出了数字签名体制的定义。

**定义 10.2 数字签名体制** 一个数字签名体制由以下部分组成:

- 一个明文消息空间  $\mathcal{M}$ : 某字母表中串的集合
- 一个签名空间  $\mathcal{S}$ : 可能的签名集合
- 一个签名密钥空间  $\mathcal{K}$ : 用于生成签名的可能密钥集合, 和一个认证密钥空间  $\mathcal{K}'$ : 用于验证签名的可能密钥集合
- 一个有效的密钥生成算法  $\text{Gen}: \mathbb{N} \mapsto \mathcal{K} \times \mathcal{K}'$ , 其中  $\mathcal{K}$  和  $\mathcal{K}'$  分别为私钥和公钥空间
- 一个有效的签名算法  $\text{Sign}: \mathcal{M} \times \mathcal{K} \mapsto \mathcal{S}$
- 一个有效的验证算法  $\text{Verify}: \mathcal{M} \times \mathcal{S} \times \mathcal{K}' \mapsto \{\text{True}, \text{False}\}$

对任意  $sk \in \mathcal{K}$  和任意的  $m \in \mathcal{M}$ , 我们用

$$s \leftarrow \text{Sign}_{sk}(m)$$

表示签名变换, 读做“ $s$  是用密钥  $sk$  生成的  $m$  的签字”。

对于任意的私钥  $sk \in \mathcal{K}$ , 用  $pk$  表示与  $sk$  相匹配的公钥。对于  $m \in \mathcal{M}$  和  $s \in \mathcal{S}$ , 必有

$$\text{Verify}_{pk}(m, s) = \begin{cases} \text{True, 概率为 } 1 & \text{若 } s \leftarrow \text{Sign}_{sk}(m) \\ \text{False, 压倒性概率} & \text{若 } s \nleftarrow \text{Sign}_{sk}(m) \end{cases}$$

其中, 概率空间包括  $\mathcal{S}, \mathcal{M}, \mathcal{K}$  和  $\mathcal{K}'$ , 如果签字/验证算法是概率算法, 那么也许还包括一个随机的输入空间。

该定义可以看做是定义 10.1 的特殊情况: 前者的  $(\text{Sign}, \text{Verify})$ 、 $(sk, pk)$  和  $(m, s)$  分别对应于后者的  $(f, g)$ 、 $(Ke, Kv)$  和  $(\text{Data}, \text{MDC})$ 。

注意, 密钥产生算法  $\text{Gen}$  的输入整数规定了输出签字/验证密钥的规模长度大小。因为密钥生成算法是有效的, 其运行时间为输入长度规模的多项式时间, 输入的整数值应该是一元编码的(原因见 4.4.6.1 节的定义 4.7)。这个整数是签名体制的安全参数, 定义了签名空间的大小。

安全参数定义了签名空间的大小, 由 4.6 节中所定义的“压倒性”概念可以知道“当  $s \nleftarrow \text{Sign}_{sk}(m)$  时, 以“压倒性”概率有  $\text{Verify}_{pk}(m, s) = \text{False}$  成立”的含意。但是, 这个概率必须忽略注释 10.1 中叙述的一种容易伪造的情况。在第 16 章研究数字签名安全性的形式化证明中, 我们会给出几种“实用”签身体制的“压倒性”的定量估计。

香农的关于加密算法混合变换的语义特性(见 7.1 节)同样适应于数字签名体制。算法  $\text{Sign}$  也应该是一个很好的混合变换函数: 输出的签名值在整个签名空间  $\mathcal{S}$  上分布得相当均匀。这个性质使得如果不使用相应的签名密钥, 就不能很容易地生成一个有效的签名。



### 10.4.1 数字签名的教科书式安全概念

我们在第8章所介绍的性质8.2(见8.2节)是基本公钥加密算法的一个教科书式的安全概念,类似于这种情况,在本章,我们也将介绍数字签名体制安全的一个非常弱的定义。

**性质 10.2 教科书式签名安全概念** 在本章范围内,我们仅考虑数字签名的一个受约束的安全概念。如果攻击者“从零开始”伪造(或生成)一个消息-签名对在计算上是不可行的,称一个数字签名是安全的。也就是说,已知公钥和关于签名体制的描述,攻击者需要输出一个有效的消息-签名对,且该消息-签名对从未被目标签名者(即已知公钥的拥有者)所签发。该攻击者是非适应的,也就是说,他不去想以某些方法使其伪造任务更容易些。例如,使用其他一些可利用的消息-签名对,或与目标签名者进行交互,使其发送供给者所选取的消息的有效签名等。

我们应当注意的是,数字签字这个安全概念对于实际应用来说是不适用的,因为它假定攻击者是无能的弱者,或其环境对攻击者是极为苛刻的。实际上,对于一个给定的公钥和签字体制,有大量的消息-签字对可以利用,因为它们并非秘密信息。况且,通常攻击者应当有权要求签名者对攻击者所选取的消息签名。这样的攻击者称为适应性攻击者,因为它可以用一种适应性方式选取消息。在这种“适应性选择消息攻击”中,给定一个目标消息,攻击者可以根据该目标消息来选择消息(可能是对目标消息做某种代数变换),并将选取的消息送给目标签名者进行签名。这好像是目标签名者为攻击者提供了签名伪造的培训课程。攻击者的任务就是伪造目标消息的签名。如同我们在8.6节所讨论的对密码系统适应性攻击的严重性一样,对签名体制的适应性选择消息攻击,尽管比非适应性攻击严重得多,但它是一种合理的攻击情况,因此应当严肃对待。

回忆我们在前面章节讨论教科书式公钥加密算法安全的教科书式定义时,多次明确地提醒公钥的拥有者不能为攻击者提供一种“天真的解密服务”。尽管要求用户保持高度警惕不是对付适应性攻击的正确解决方案,但是,如果密钥的拥有者足够聪明的话,这种程度的警惕是可以做到的。现在,在数字签名中,我们可以不再要求或提醒用户不要提供“天真的签名服务”。签名服务是不可避免的:在许多应用中,对所给消息进行签字是一种完全正常的服务。

数字签名的一个强安全概念,称为抵抗适应性选择消息攻击的不可伪造性,是数字签名的一种适于应用的安全概念,作为密码系统中的CCA2(见8.6节的定义8.3)类似情况将在第16章介绍。在那里,依据这种强安全概念,我们还将研究一些数字签字体制的形式化安全性论证。

我们对一种容易而又无大碍的签名伪造形式注释如下:

**注释 10.1 存在性伪造** 算法 $(\text{Sign}_{sk}, \text{Verify}_{pk})$ 构成一个单向陷门函数对,其中单向部分是 $\text{Verify}_{pk}$ ,陷门部分是 $\text{Sign}_{sk}$ 。通常,函数 $\text{Verify}_{pk}(s, m)$ 是从 $s$ 到 $m$ 方向计算的。因此,许多基于单向陷门函数的数字签名体制一般都提供一个有效的方法,利用从 $s$ 到 $m$ 计算的单向函数 $\text{Verify}_{pk}$ 来伪造“有效的消息-签名”对。但是,考虑到由于单向函数 $\text{Verify}_{pk}$ 也必须具有混合变换性质,所以利用函数 $\text{Verify}_{pk}$ 从一个“签名”所生成的“消息”将是随机的,几乎一定是没有意义的。这种简单的伪造方法就是伪造技术中一个组成部分,称之为存在性伪造。基于单向陷门函数的数字签名体制一般允许这种伪造。为防范存在性伪

造,通常的方法是为所要签名的消息增加一些可辨认的冗余,使得验证者可以验证消息的非随机分布性。□

现在,让我们介绍几个著名的数字签名体制。

### 10.4.2 RSA 签字体制(教科书式版本)

RSA 签名体制是继 Diffie 和 Hellman 提出数字签名思想后的第一个数字签名体制,它是由 Rivest、Shamir 和 Adleman 三人实现的[248]。RSA 签名体制的详细说明见算法 10.1。我们注意到,这是 RSA 签名体制算法的教科书式版本。

#### 算法 10.1 RSA 签名体制

##### 密钥建立

密钥建立过程和 RSA 密码系统的密钥建立过程(算法 8.1)相同。

(\* 因此,用户 Alice 的公钥为  $(N, e)$ , 其中  $N = pq$ ,  $p$  和  $q$  是两个长度差不多的大素数,  $e$  是满足  $\gcd(e, \phi(N)) = 1$  的整数。她也可以找到一个整数  $d$ , 满足  $ed \equiv 1 \pmod{\phi(N)}$ 。整数  $d$  就是 Alice 的私钥。\*)

##### 签名生成

为了生成消息  $m \in \mathbb{Z}_N^*$  的签名, Alice 生成

$$s = \text{Sign}_d(m) \leftarrow m^d \pmod{N}$$

##### 签名验证

设 Bob 是验证者,他知道公钥  $(N, e)$  属于 Alice。给定一个消息-签名对  $(m, s)$ , Bob 的验证过程为

$$\text{Verify}_{(N, e)}(m, s) = \text{True}, \text{ 其时 } m \equiv s^e \pmod{N}$$

(\* 注意,  $m$  必须是一个可识别的消息,见 10.4.3 节。\*)

很容易看出, RSA 数字签名过程与 RSA 加密和解密过程(见 8.5 节)的格式相同,惟一不同的是,现在 Alice 首先用她的私钥进行“加密”,而 Bob(或任何人)再用 Alice 的公钥进行“解密”。有效签名验证一致性的成立恰好由 8.5 节中的 RSA 加密和解密一致性论证而来。

### 10.4.3 RSA 签字安全性的非形式化论证

如果 RSA 签字体制就像我们所描述的那样简单,那么任何人要伪造 Alice 的签名都不难。例如, Bob 可以选取一个随机数  $s \in \mathbb{Z}_N^*$ , 并计算

$$m \leftarrow s^e \pmod{N} \quad (10.4.1)$$

当然,对这样事先准备好的“消息”-签名对,其验证结果回报为 True。而且, RSA 函数的乘法性质(见 8.9 节的式(8.9.1))提供了一个简单的方法,从已知的消息-签名对伪造新的消息-签名对,例如从现有的消息-签名对  $(m_1, s_1)$  和  $(m_2, s_2)$  伪造一个新的消息-签名对  $(m_1 m_2, s_1 s_2)$ 。

就像我们在注释 10.1 中所说的,上面的伪造方法是存在性伪造。因为在式(10.4.1)中或通过相乘所生成的  $m$  看起来应该是随机的,因此常通过为  $m$  增加一些可识别的冗余信息,使

之变得不随机或“是有意义的”来抗击这种存在性伪造。为消息增加可识别信息的最简单方法是使消息本身包含可识别的部分。例如  $m = M \parallel I$ , 其中  $M$  是真正要签名的消息,  $I$  为可识别的串, 比如签名者的身份。

为消息增加可识别信息的最常用方法是利用密码杂凑函数(见 10.3.1 节)对该消息进行“杂凑”。设  $h$  为从  $\{0,1\}^*$  映射到  $\mathcal{M}$  的杂凑函数, 如果存在位串  $M \in \{0,1\}^*$  和消息  $m \in \mathcal{M}$ , 并满足

$$m = h(M)$$

那么就认为消息  $m \in \mathcal{M}$  是可识别的或有意义的。

在这种消息可识别性的概念下, 伪造 RSA 签名不再是一件容易的事。如果攻击者不能给出可识别的消息  $m$ , 例如, 攻击者知道所用密码杂凑函数下  $m$  的原像, 那么像式 (10.4.1) 那样从  $s$  计算  $m$  就不再是有用的伪造方法。如果我们假设杂凑函数具有类似于随机预言机的行为(随机预言机的行为在 10.3.1.2 节中做了描述), 那么“从零开始的伪造”给定消息的 RSA 签名应该具有解 RSA 问题的困难性, 也就是说, 求取模  $N$  下的  $e$  次根的困难性(见 8.7 节中定义 8.4)。

但必须注意的是, 我们没有给出这一结果的任何形式化证据(即证明)。算法 10.1 所述的教科书式 RSA 签名体制肯定不具有可证明安全性。对这个使用消息杂凑函数的简单版本, 在适应性选择消息攻击模式下, 没有人知道如何证明它的安全性。因此, 这个简单的版本也应当列为教科书式 RSA 签名。

使用杂凑函数的 RSA 签名的一个更好的算法将在第 16 章介绍。这个算法是一个概率算法, 意味着该签名算法的签名输出在签名空间上随机分布, 与均匀分布是不可区分的。该算法也是 RSA 签名体制的一种适于应用的版本。该 RSA 签名体制安全性的形式化论证将会在一个更强的、适于应用的安全性概念下讨论, 此概念也将在第 16 章中介绍。

#### 10.4.4 Rabin 签名体制(教科书式版本)

Rabin 签名体制[242]与 RSA 签身体制很相似, 所不同的是它们使用不同的验证指数。在 RSA 签字中, 验证指数  $e$  是一个奇整数, 因为它要求  $\gcd(e, \phi(N)) = 1$ , 其中  $\phi(N)$  是一个偶数; 而在 Rabin 签字中,  $e = 2$ 。

对 RSA 签名体制的详细说明见算法 10.2。我们注意到, 这是 Rabin 签身体制算法的教科书版本。

与 RSA 签名相比, Rabin 签名具有两个优点。一是可以证明伪造与分解问题同等困难(形式化的证明放在后面进行); 二是验证速度更快, 适合于采用小型计算设备进行签字验证的应用场合, 如掌上设备等。

由注释 10.1 有, 如果  $m$  不是一条可识别的消息, 那么对于 Rabin 签名体制来说, 伪造一个有效的“消息”-签名对是很容易的。这就是存在性伪造。常用的防范方法如 10.4.3 节中所述的那样, 对消息进行杂凑, 使消息可以识别。

#### 10.4.5 关于 Rabin 签名的一个自相矛盾的安全性基础

依据与定理 8.2(见 8.11 节)所述的相同思想, 我们也可以证明, 如果存在一个可以伪造 Rabin 签名的算法, 那么该伪造算法可用于分解签名体制中所用的合数模。这是一个令人满意的性质, 因为它将伪造签名和一个著名的困难问题(分解)联系起来。

但是,这种强安全性质也意味着,对于适应性攻击,Rabin 签字体制是极其不安全的。在适应性攻击中,攻击者可以要求签名者发布对攻击者任意选取的消息的签名。例如,攻击者可以选取任意的  $s \in \mathbb{Z}_N^*$ ,并将  $m = s^2 \pmod{N}$  递交给 Alice,要求她返回消息  $m$  的 Rabin 签名。Alice 的应答,记为  $s'$ ,是  $m$  的四个平方根中的任意一个。如果  $s' \not\equiv \pm s \pmod{N}$ ,那么适应性攻击者就可以分解她的模数。

因此,算法 10.2 所给定的教科书式 Rabin 签字体制在任何实际的应用环境中是绝对不可用的,其中的适应性攻击是不可避免的。实际应用中的 Rabin 签名必须防范适应性攻击者得到一条消息的两个不同的平方根。

### 算法 10.2 Rabin 签名体制

#### 密钥建立

用户 Alice 建立如 RSA 模那样的公共模。

(\* 因此,她的模为  $N = pq$ ,其中  $p$  和  $q$  是两个不同的奇素数。 $N$  是她的公钥, $p$  和  $q$  是它的私钥。\*)

#### 签名生成

为了生成消息  $m \in \mathbb{Z}_N^*$  的签名。Alice 生成签名如下

$$s \leftarrow m^{1/2} \pmod{n}$$

(\* 为了使这个计算可行,必须使  $m \in \text{QR}_N$ ;根据 6.6.2 节,我们可以知道,对于 RSA 模  $N$  有  $\#\text{QR}_N = \#\mathbb{Z}_N^*/4$  成立,也就是说  $\mathbb{Z}_N^*$  中四分之一的元素在  $\text{QR}_N$  中;因此,Alice 可以使用一种合适的消息格式化机制,以便她可以确信  $m \in \text{QR}_N$ ;对于这样的  $m$ ,Alice 可以运用算法 6.5 计算它的一个平方根。\*)

#### 签名验证

设 Bob 是一个验证者,他知道公共模  $N$  属于 Alice。给定一个消息 - 签名对  $(m, s)$ ,Bob 的验证过程为

$$\text{Verify}_N(m, s) = \text{True}, \text{ 其时 } m \equiv s^2 \pmod{N}$$

(\* 注意, $m$  必须是一个可识别的消息,见 10.4.3 节。\*)

在第 16 章将介绍采用杂凑函数的一个更好的、适合应用 Rabin 签名方案。该算法是一个概率算法,确保了对同一消息所发布的多次签字是随机的,这使得适应性攻击者不能得到一条消息的两个不同的平方根。因此,该 Rabin 签名方案是一个适合应用的方案。Rabin 签名体制安全性的形式化论证将在一种更强的、适于应用的安全性概念下考虑,我们也放在第 16 章中介绍。

关于 Rabin 签名体制的安全性,我们总结为一个自相矛盾的结果。

一方面,使用与定理 8.2(见 8.11 节)所采用的相同的方法,可以证明对于教科书式 Rabin 签名,不可伪造性的教科书式意义等同于分解问题。这不仅是一个非常强的结果,因为它具有形式化的证据(即证明);而且这这也是一个令人满意的结果,因为它将签名伪造和一个著名的困难问题联系起来:整数分解。

另一方面, Rabin 签名体制的这个教科书式版本的安全性又非常弱, 在实际的应用环境中是绝对不可用的, 在实际应用中适应性选择消息攻击是很常见的。这种攻击完全破坏了 Rabin 签名体制。我们需要一种适于应用的 Rabin 签名体制的变型, 第 16 章将介绍这种变型。然而遗憾的是, 当我们看到这个变型体制时, 我们就会发现, 该体制安全性证明(安全性的形式化证据)不再将不可伪造性和整数分解联系起来。

### 10.4.6 ElGamal 签名体制

除了在 8.12 节中所述的完美公钥密码系统外, ElGamal 还设计了一个精巧的数字签名体制。像 ElGamal 公钥密码系统一样, 激发了随之而来的应用研究领域极大的兴趣, 一直延续到今天。ElGamal 签名体制也是许多其他数字签名体制的起源, 这些签名体制都属于类 ElGamal 签名体制族(其中一些将在 10.4.8 节中介绍, 而它们的安全性将在第 16 章进一步研究)。

ElGamal 签名体制的详细说明见算法 10.3。

#### 算法 10.3 ElGamal 签名体制

##### 密钥建立

密钥建立过程和 ElGamal 密码系统的密钥建立过程(算法 8.12)相同。

(\* 因此, 用户 Alice 的公钥为三元组  $(g, y, p)$ , 其中  $p$  是一个大的素数,  $g \in \mathbb{F}_p^*$  是一个随机的乘法生成元, 并且对于某个秘密整数  $x_A < p - 1$  有  $y_A \equiv g^{x_A} \pmod{p}$ ; Alice 的私钥为  $x_A$ 。\*)

##### 签名生成

为了生成消息  $m \in \mathbb{F}_p^*$  的签名, Alice 选取一个随机数  $\ell \in {}_U\mathbb{Z}_{p-1}^*$  (即  $\ell < p - 1$  且  $\gcd(\ell, p - 1) = 1$ ), 并生成一个签名对  $(r, s)$ , 其中

$$\begin{aligned} r &\leftarrow g^\ell \pmod{p} \\ s &\leftarrow \ell^{-1}(m - x_A r) \pmod{p-1} \end{aligned} \quad (10.4.2)$$

(\*  $\ell^{-1}$  可用扩展的欧几里得算法(算法 4.2)计算。\*)

##### 签名验证

设 Bob 是一个验证者, 他知道公钥  $(g, y_A, p)$  属于 Alice。给定一个消息 - 签名对  $(m, (r, s))$ , Bob 的验证过程为

$$\text{Verify}_{(g, y_A, p)}(m, (r, s)) = \text{True}$$

$$r < p \text{ 和 } y_A r^s \equiv g^m \pmod{p}$$

(\* 注意,  $m$  必须是一个可识别的消息, 见 10.4.7.2 节。\*)

### 10.4.7 ElGamal 签名体制安全性的非形式化论证

我们现在来研究有关 ElGamal 签名体制安全性的几个的结果。

#### 10.4.7.1 提醒

在 ElGamal 签名体制中, 我们应该注意以下几个提醒。

**提醒 1**

第一个提醒是在签名验证过程中,检验  $r < p$  的重要性。Bleichenbacher[42]发现了下面的攻击,条件是在 Bob 接受的签名中有  $r > p$  成立。设  $(r, s)$  表示消息  $m$  的签名, Malice 可以通过下面的方法伪造任意一个消息  $m'$  的一个新签名:

1.  $u \leftarrow m' m^{-1} \pmod{p-1}$
2.  $s' \leftarrow su \pmod{p-1}$
3. 计算  $r'$ , 使之满足  $r' \equiv ru \pmod{p-1}$  和  $r' \equiv r \pmod{p}$ ; 这可以应用中国剩余定理来完成(算法 6.1)。

然后,例行的做法是检验下面的同余等式成立:

$$y_A^{r'} r'^{s'} \equiv y_A^m r^{su} \equiv (y_A^r r^s)^u \equiv g^{mu} \equiv g^{m'} \pmod{p}$$

如果 Bob 对  $r < p$  进行检验,就会阻止这样的攻击。这是因为在上面第 3 步中由中国剩余定理所计算的  $r'$  是一个  $p(p-1)$  量级的量。

**提醒 2**

第二个提醒也是由 Bleichenbacher[42]发现的: Alice 应该从  $\mathbb{F}_p^*$  中随机地选取公开参数  $g$ 。如果该参数不是由 Alice 选取的(例如,系统中的所有用户共享相同的公开参数  $g, p$ ),那么必须有一个所有用户都知道的公共过程来检验  $g$  选择的随机性(例如,  $g$  是一个伪随机函数的输出)。

现在,我们假设公开参数  $g, p$  是由 Malice 选择的。参数  $p$  可以通过我们在 8.4.1 节中所推荐的标准方法来建立: 设  $p-1 = bq$ , 其中  $q$  可以是一个足够大的素数,而  $b$  可以是平滑的(也就是说,  $b$  仅有一个小的素因子,因此在  $b$  阶群上计算离散对数是容易的,见 8.4.1 节)。

Malice 可以如下产生  $g$ :

$$g = \beta^c \pmod{p}$$

对某个  $\beta = cq$ , 有  $c < b$ 。

对 Alice 的公钥  $y_A$ , 我们知道,当底为  $g$  时,求  $y_A$  的离散对数是困难的。但是,当底为  $g^q$  时,求取  $y_A^q$  的离散对数是容易的。该离散对数为  $z \equiv x_A \pmod{b}$ , 也就是说,下面的同余等式成立:

$$y_A^q \equiv (g^q)^z \pmod{p}$$

有了  $z$ , Malice 可以伪造 Alice 的签名如下:

$$\begin{aligned} r &\leftarrow \beta = cq \\ s &\leftarrow t(m - cqz) \pmod{p-1} \end{aligned}$$

然后,例行的做法是检验下面的同余等式成立:

$$y_A^r r^s \equiv y_A^{cq} (\beta^c)^{(m - cqz)} \equiv g^{cqz} g^{m - cqz} \equiv g^m \pmod{p}$$

因此,  $(r, s)$  实际上是  $m$  的一个有效签名,其生成过程没有使用  $x_A$  而是使用  $x_A \pmod{b}$ 。

我们注意到,在这个签名伪造攻击中,  $r$  是一个可以被  $q$  整除的值。而在  $p$  的标准参数建立阶段,  $p$  满足  $p = bq$ , 其中  $q$  是一个大的素数。因此,在验证过程中,如果 Bob 验证  $(q \nmid r)$ , 那么就可以防范 Bleichenbacher 所发现的攻击(假定在  $p$  的标准设定阶段使得  $q$  成为公钥参数



的一部分)。关于这一点,在后面的 16.3.2.1 节中对 ElGamal 签名体制的不可伪造性进行形式化证明时,我们将看到,为了完成形式化证明,必须有条件( $q \nmid r$ )。

### 提醒 3

第三个提醒是关于短暂密钥  $\ell$ 。类似于 ElGamal 加密:ElGamal 签名的生成过程也是一个随机的算法。它的随机性是由于这个短暂密钥  $\ell$  的随机性。

Alice 永远不要在不同的签名过程中重复使用同一个短暂密钥。如果重复使用一个短暂密钥  $\ell$  对两个消息  $m_1 \neq m_2 \pmod{p-1}$  进行签字,那么由式(10.4.2)中的第二个等式,我们有

$$\ell(s_1 - s_2) \equiv m_1 - m_2 \pmod{p-1}$$

因为  $\ell^{-1} \pmod{p-1}$  存在,  $m_1 \neq m_2 \pmod{p-1}$  意味着

$$\ell^{-1} \equiv (s_1 - s_2) / (m_1 - m_2) \pmod{p-1} \quad (10.4.3)$$

即得到了  $\ell^{-1}$ 。依次地,可以从式(10.4.2)中的第二个等式计算出 Alice 的私钥  $x_A$ :

$$x_A \equiv (m_1 - \ell s_1) / r \pmod{p-1} \quad (10.4.4)$$

还应该注意,这个短暂的密钥必须从空间  $\mathbb{Z}_{p-1}^*$  中随机均匀地选取。当一个签名是由小型计算机生成时,比如智能卡或掌上设备等,应该特别注意:必须要确保这些设备配备了足够的、可依赖的随机资源。

只要保证  $\ell$  在每次签名中仅使用一次,并且它是随机均匀产生的,那么签名过程(10.4.2)的第二个等式说明,它实质上是为签名者的私钥  $x$  提供了一次性的乘法加密。因此,这两个密钥在信息论安全的意义上彼此互相保护。

#### 10.4.7.2 存在性伪造的防范

如果所要签名的消息不包含可识别的冗余,那么注释 10.1 中所述的存在性伪造也适于 ElGamal 签名。也就是说,如果所要签名的“消息”不是一个可识别的消息,在 ElGamal 签名体制下,伪造一个有效的“消息”-签名对不是一件困难的事。

例如,令  $u, v$  是小于  $p-1$  的任意整数,且满足  $\gcd(v, p-1) = 1$ ; 设

$$r \leftarrow g^u y_A^v \pmod{p}$$

$$s \leftarrow -rv^{-1} \pmod{p-1}$$

$$m \leftarrow -ruw^{-1} \pmod{p-1}$$

那么  $(m, (r, s))$  确实就是在 ElGamal 签名体制下与 Alice 的公钥  $y_A$  有关的一个有效的“消息”-签名对。这是因为

$$\begin{aligned} y^r r^s &\equiv y_A^r r^{-n^{-1}} \\ &\equiv y_A^r (g^u y_A^v)^{-n^{-1}} \\ &\equiv y_A^r (g^u)^{-n^{-1}} (y_A^v)^{-n^{-1}} \\ &\equiv y_A^r g^{-nu^{-1}} y_A^{-r} \\ &\equiv g^{-nu^{-1}} \\ &\equiv g^m \pmod{p} \end{aligned}$$

然而,在这个伪造中,由于模指数具有好的混合变换性质,“消息” $m$ 是不可识别的。

消息格式化机制可以挫败这种伪造。一个最简单的消息格式化机制就是使 $m$ 含有一个可识别的部分,例如, $m = M \parallel I$ ,其中 $M$ 是所要签名的消息, $I$ 为一个可识别的串,比如签名者的身份。

最常用的消息格式化机制是使 $m$ 为所要签名的消息的杂凑值。一个杂凑消息的例子可为

$$m = H(M, r)$$

这里, $H$ 是一个密码杂凑函数, $M$ 是某一条消息的比特串。现在,所得的签名就是对消息 $M$ 的签名。验证步骤包括对 $m = H(M, r)$ 的验证。杂凑函数所具有的单向性有效地阻止了前面所说的存在性伪造。

如果我们假设杂凑函数 $H$ 的行为如同一个随机预言机(见 10.3.1.2 节),那么就可以得到形式化的证据,将 ElGamal 签名的不可伪造性和离散对数问题(著名的困难问题)联系起来。但现在,我们还没有足够的工具来论证这些形式化的证据。形式化的证明将推迟到第 16 章。

同样的道理,我们也将 ElGamal 签名族中其他签名体制安全性的形式化证明推迟到第 16 章。

#### 10.4.8 ElGamal 签名族中的签名体制

在 ElGamal 的开创性工作之后,出现了几个 ElGamal 签名体制的变型。最有影响的两个变型是 Schnorr 签名体制[258,259]和美国数字签名标准(DSS)[217,218]。

##### 10.4.8.1 Schnorr 签名体制

Schnorr 签名体制是 ElGamal 签名体制的一个变型,但它本身还具有一个特点:可以大大地缩短素数域元素的表示,而不降低困难问题(DL 问题,见 8.4 节)的难度。这是对公钥密码学的一个重要贡献。这个思想后来发展成为更一般形式有限域的一种新的密码系统:XTR 公钥系统[177]。

可以通过构造一个域 $\mathbb{F}_p$ ,使之包含一个更小的、素数阶 $q$ 的子群,来实现缩短表示。我们注意到,在类 ElGamal 密码系统中,目前参数 $p$ 的标准设置为 $p \approx 2^{1024}$ 。我们应当进一步注意,为了适应在解 DL 问题方面的进步, $p$ 的长度很可能需要增大。但是,在 Schnorr 的工作之后,将参数 $q$ 设置为 $q \approx 2^{160}$ 已成为一种标准的约定(经验规则)。这种设置很可能或多或少地保持不变,而不考虑 $p$ 的长度的增加。这是因为子群的信息对于解 $\mathbb{F}_p$ 上 DL 问题的一般方法来说不会起到什么作用,即使知道了目标元素在给定的子群中也没用。设定 $q$ 为恒定的 $2^{160}$ 仅仅是由平方根攻击(见 3.6 节)的下界要求定出的。

Schnorr 签名体制的详细说明见算法 10.4。

注意,在公开参数设置过程中,可很快找到一个生成元 $g$ 。这是因为对 $q \mid p-1$ ,有

$$\text{Prob}[\gcd(\text{ord}(f), q) = 1 \mid f \in {}_U\mathbb{Z}_p^*] \leq 1/q$$

也就是说,随机选取的函数 $f$ 满足 $g \leftarrow f^{\frac{p-1}{q}} \equiv 1 \pmod{p}$ 的概率很小,可以忽略。根据费马小定理(见 6.4 节中的定理 6.10),我们有

$$g^q \equiv 1 \pmod{q}$$

因此, $g$ 实际上生成了一个含有 $q$ 个元素的子群。

### 算法 10.4 Schnorr 签名体制

#### 系统参数的建立

1. 选取两个素数  $p$  和  $q$ , 使其满足  $q \mid p-1$ ;  
(\* 这两个参数的典型长度分别为  $|p| = 1024$  和  $|q| = 160$  \*)
2. 选取一个  $q$  阶元素  $g \in \mathbb{Z}_p^*$ ;  
(\* 其方法可以为选取一个  $f \in {}_U \mathbb{Z}_p^*$  并设置  $g \leftarrow f^{(p-1)/q} \pmod{p}$ 。如果  $g = 1$ , 重复这个过程直至  $g \neq 1$  \*)
3. 建立一个密码杂凑函数  $H: \{0, 1\}^* \mapsto \mathbb{Z}_p$ ;  
(\* 例如, SHA-1 是  $H$  的一个很好的候选函数 \*)

#### 主体公/私钥的建立

用户 Alice 选取一个随机数  $x \in {}_U \mathbb{Z}_p^*$  并计算

$$y \leftarrow g^{-x} \pmod{p}$$

Alice 的公钥为  $(p, q, g, y, H)$ ; 她的私钥为  $x$ 。

#### 签名生成

为了生成消息  $m \in \{0, 1\}^*$  的签名, Alice 选取一个随机数  $\ell \in {}_U \mathbb{Z}_p^*$ , 并生成一个签名对  $(e, s)$ , 其中

$$\begin{aligned} r &\leftarrow g^\ell \pmod{p}; \\ e &\leftarrow H(m \parallel r); \\ s &\leftarrow \ell + xe \pmod{q} \end{aligned}$$

#### 签名验证

设 Bob 是一个验证者, 他知道公钥  $(p, q, g, y, H)$  属于 Alice。给定一个消息 - 签字对  $(m, (e, s))$ , Bob 的验证过程为

$$\begin{aligned} r' &\leftarrow g^s y^e \pmod{p} \\ e' &\leftarrow H(m \parallel r') \\ \text{Verify}_{(p, q, g, y, H)}(m, (s, e)) &= \text{True}, e' = e \end{aligned}$$

签名验证过程是正确的, 这是因为如果  $(m, (s, e))$  是 Alice 所生成的有效的消息 - 签名对, 那么有

$$r' \equiv g^s y^e \equiv g^{s+e\ell} y^e \equiv y^{-e} g^\ell y^e \equiv g^\ell \equiv r \pmod{p}$$

像我们在前面所讨论的, 在  $\mathbb{F}_p$  的  $q$  阶子群中, Schnorr 签名体制的签名要比 ElGamal 签名体制的签名短得多: 传送一个 Schnorr 签名需要  $2|q|$  比特, 而传送一个 ElGamal 签名需要  $2|p|$  比特。短的签名也意味着签名生成和验证过程所需的操作更少: Schnorr 签名为  $O_B(\log_2 q \log^2 p)$ , 而 ElGamal 签名为  $O_B(\log^3 p)$ 。需进一步注意的是, 在签名生成过程中, 模  $p$  部分的计算可以以离线的方式完成。考虑到这一点, 实时的签名生成仅需要计算一次模  $q$  乘法, 其他的艰苦工作离线完成。这样的设计安排适合于小型的运行设备。

与 ElGamal 签名的情况相同,永远不要重复使用短暂密钥 $\ell$ ,并且 $\ell$ 应该是均匀随机的。在这种条件下,短暂密钥和签名者的私钥在信息论安全的意义上互相保护。

#### 10.4.8.2 数字签名标准(DSS)

1991 年 8 月,美国标准化组织,美国国家标准技术研究所(NIST)公布了一种新的、建议采用的数字签名体制,称为数字签名标准(DSS)[217,218]。DSS 本质上是 ElGamal 签名体制,但它和 Schnorr 签名体制一样,也是运行在较大有限域中一个很小的素阶子群内,在这个有限域中,DL 问题是困难问题。因此,比起 ElGamal 签名体制,DSS 大大地减小了签名的长度。

DSS 的详细说明见算法 10.5。

#### 算法 10.5 数字签名标准

##### 系统参数的建立

(\* 系统参数的建立过程和 Schnorr 签名体制相同;因此,公开参数 $(p, q, g, H)$ 以便系统中所有用户使用,参数的意义与算法 10.4 中的相同。\*)

##### 主体公/私钥的建立

用户 Alice 选取一个随机数  $x \in_U \mathbb{Z}_q$  作为它的私钥,并通过下面的公式计算她的私钥

$$y \leftarrow g^x \pmod{p}$$

Alice 的公钥为  $(p, q, g, y, H)$ ; 她的私钥为  $x$ 。

##### 签名生成

为了生成消息  $m \in \{0, 1\}^*$  的签名, Alice 选取一个随机数  $\ell \in_U \mathbb{Z}_q$ , 并生成一个签名对  $(r, s)$ , 其中

$$\begin{aligned} r &\leftarrow (g^\ell \pmod{p}) \pmod{q} \\ s &\leftarrow \ell^{-1} (H(m) + xr) \pmod{q} \end{aligned}$$

##### 签名验证

设 Bob 是一个验证者,他知道公钥  $(p, q, g, y, H)$  属于 Alice。给定一个消息-签名对  $(m, (r, s))$ , Bob 的验证过程为

$$\begin{aligned} w &\leftarrow s^{-1} \pmod{q} \\ u_1 &\leftarrow H(m)w \pmod{q} \\ u_2 &\leftarrow rw \pmod{q} \\ \text{Verify}_{(p, q, g, y, h)}(m, (r, s)) &= \text{True}, r(g^{u_1} y^{u_2} \pmod{p}) \pmod{q} \end{aligned}$$

签名验证过程是正确的,这是因为如果  $(m, (r, s))$  是由 Alice 生成的有效消息-签名对,则有

$$y^{u_1} y^{u_2} \equiv g^{H(m)s^{-1}} y^{rs^{-1}} \equiv g^{(H(m)+x)s^{-1}} \equiv g^\ell \pmod{p}$$

把右边的项同签名生成过程的第一个等式做比较可知,如果对其进一步做模  $q$  运算,就会到得  $r$ 。

如果公开参数长度相同,DSS 所需通信带宽和计算要求与 Schnorr 签名体制的相同。

DSS 和所采用的杂凑函数,即 SHA-1[219],用一种可兼容的标准过程结合成一体被定为标准。标准杂凑函数的使用为消息的可识别性提供了所需的性质,从而阻止了存在性伪造。

最后,同 ElGamal 签名族中所有的签名体制一样,DSS 也必须注意短暂密钥的问题。

#### 10.4.9 数字签名体制安全性的形式化证明

类似于 8.14 节中关于公钥密码系统对强安全性概念的需要进行讨论,我们也应该对数字签名体制可证明安全性的问题进行简短的讨论。

读者可能已经注意到,我们没有对本章所介绍的数字签名体制的安全性给出任何形式化证明。实际上,我们在注释 10.2 中说过,本章不考虑签名体制的形式化证明。对此,有两个原因。

为了解释第一个原因,我们注意到,“从零开始”伪造一个签名应比利用一些已有的消息-签名对来伪造一条消息的签名困难,这样的期望是很合理的。攻击者可能在开始伪造签名之前就已经获得了这些消息-签名对。如果攻击者可以和目标签名者进行交互,并劝说他提供签名服务,也就是发布为攻击者选择的消息所进行签名,那么伪造签名的任务就会变得更为容易。基于目标签名者签名服务的签名伪造称为通过**适应性选择消息攻击**的伪造。

实际上,可以得到关于给定公钥的大量的消息-签名对。而且,在数字签名的应用中,适应性攻击是很难防范的:在许多应用中,为给定消息发布签名是完全合法的服务。因此,必须给出数字签名安全性的适于应用的概念。我们将在第 16 章定义这样的安全性概念。这是将数字签名安全性的形式化证明放在后面章节的第一个原因。

对于第二个原因,我们已经知道,如果一条“消息”是不可识别的(一般性,见注释 10.1 关于存在性伪造的容易性;特定的,可以回顾我们在描述各种具体体制时给出的存在性伪造的许多具体情况),那么伪造一条消息-签名对一般是很容易的,即使是“从零开始”进行伪造也都不难。为了防范这些容易的伪造方法,任何数字签名体制都必须配置消息格式化机制,将所要签名的消息转换成可识别的消息。消息格式化机制通常用密码杂凑函数实现。因此,为了形式化证明数字签名体制的安全性,假设有一个密码杂凑函数模型化的行为是很合理的。由于缺少这样的模型化行为,我们还不能给出本章目前所介绍的数字签名体制安全性的形式化论证。这就是将数字签名安全性的形式化证明放在后面章节的第二个原因。

我们在 10.3.1.2 节已经讨论过密码杂凑函数试图仿真随机函数。对于使用杂凑函数的密码体制,一个用来形式化证明它的安全性的概念称为**可证明安全的随机预言机模型(ROM)**。这个概念将在第 16 章介绍。在那里将会看到,利用 ROM,我们能够给出形式化证明,将伪造签名的困难性(甚至通过适应性选择消息攻击)和计算复杂度理论中的一些著名的计算假设联系起来。

### 10.5 非对称技术Ⅱ:无源识别的数据完整性

在利用数字签名体制实现的数据完整性机制中,密钥参数通常的设置约定为: $K_e$  为私钥, $K_d$  为与之匹配的公钥。在这种情况下,一条消息完整性的正确验证结果向验证者提供了有关消息发送者,即该消息的签名者,亦即公钥  $K_d$  的拥有者的身份。

然而,我们应该注意,尽管这种“密钥参数常用的设置方式”是实现数字签名体制的一个必要组成部分,但是对于数据完整性系统并不是必需的。事实上,在定义 10.1 中,我们从未对生成和验证 MDC 的两个密钥做任何限制。

因此,我们确实可以设置这两个密钥  $K_e$  和  $K_v$ ,比如与数字签名体制相反的方式,也就是说,设  $K_e$  为公钥,而  $K_v$  为私钥。在这种密钥设置方式下,任何人都可以使用公钥  $K_e$  生成一个一致性的(即密码学完整的)数值对(Data, MDC)或“消息-签名对”(m, s),只有私钥  $K_v$  的拥有者才可以验证数值对(Data, MDC)的一致性,或“签名”(m, s)的有效性。当然,在这种不常见的密钥设置方式下,这样的系统不能再看做是数字签名体制。但是,我们必须注意,由定义 10.1 可知,在这种不常见的密钥设置方式下,这个系统仍然是一个数据完整性系统!

因为任何人都可以使用公钥  $K_e$  生成一个一致性的数值对(Data, MDC),我们称这种数据完整性系统为**无源识别的数据完整性**。依据我们对 Malice(坏人)的行为的了解,为了方便,将这种数据完整性服务另称之为“来自 Malice 的数据完整性”,不会给我们带来危害。

现在让我们来看一个提供这种服务的公钥加密体制的例子。该体制具有这样一个性质: Malice 可以送给 Alice 一条秘密的消息,且该消息是“不可展的”(比如, Malice 的其他朋友不能修改),也就是说,对于 Malice 所在团伙中的任何其他的人来说,要修改该消息而不被消息的接收者 Alice 发现在计算上是困难的。该算法的 RSA 示例在算法 10.6 中详细说明,这个算法称为**最优非对称加密填充(OAEP)**,是由 Bellare 和 Rogaway[25]两人设计的。

如果密文发出后没有被修改过,那么我们知道, Alice 可以用解密算法正确地恢复出随机数  $r$ ,因此有

$$v = s \oplus G(r) = (m \parallel 0^{k_1}) \oplus G(r) \oplus G(r) = m \parallel 0^{k_1}$$

所以, Alice 将看到所恢复的明文消息后面是  $k_1$  个零。

另一方面,对密文的任何修改都会导致 RSA 函数所密封的消息的改变。这种改变将进一步导致对明文消息(包括输入到 OAEP 函数的随机输入以及明文消息尾部  $k_1$  个零的冗余,的“不可控制的”改变。从直觉上看,这种“不可控制的”改变是由该体制所用两个杂凑函数的所谓的“随机预言机”性质引起的(见 10.3.1.2 节关于随机预言机的讨论)。这种不可控制的改变的出现是由于破坏了增加到明文消息中的冗余( $k_1$  个零的串),其概率至少为  $1 - 2^{-k_1}$ 。已知  $2^{-k_1}$  是可忽略的,所以  $1 - 2^{-k_1}$  就接近于 1。因此,这个体制的确为加密的消息提供了一种数据完整性保护。

注意,由 RSA-OAEP 加密算法提供的数据完整性保护很奇特:尽管 Alice 在看到  $k_1$  个零的串之后能够确信密文没有被修改过,但是她却不可能知道该消息的发送者是谁。这就是在算法 10.6 中,我们故意将 Malice 指定为发送者的原因。这个“来自 Malice 的数据完整性”的概念非常有用,也非常重要。很明显,它是抗击适应性选择密文攻击(CCA2, 见 8.6 节中的定义 8.3)的公钥加密体制发展的结果。在抗击 CCA2 的公钥密码系统中,解密过程包括数据完整性验证。这样的密码系统被认为是无懈可击的,即使对下述的被攻击者滥用的极端形式:

- 攻击者和公钥的拥有者进行询问-应答游戏。攻击者处于询问者的位置,他可以自由地向公钥的拥有者发送任意多的(当然,攻击者是多项式限制的)“适应性选择密文”消息,并要求他以预言机-服务的方式进行解密(回顾我们在 8.2 节关于“预言机服务”的讨论,并看一下 8.2 节中给出的预言机解密服务的具体实例)。
- 公钥的拥有者处于应答者的位置。如果在解密过程中通过了数据完整性验证,那么公钥的拥有者应当简单地送回解密的结果,而忽略该解密请求可能来自于一个攻击者的事实。这个攻击者可能是以某种巧妙的、不公开的方式生成了该密文,其目的是为了攻



破该目标密码系统(或者获得攻击者无权看到的明文消息,或者获得密钥拥有者的私钥)。

#### 算法 10.6 RSA 最优非对称加密填充(RSA-OAEP)[25]

##### 密钥参数

设  $(N, e, d, G, H, n, k_0, k_1) \leftarrow_U \text{Gen}(1^k)$  满足:  $(N, e, d)$  是 RSA 的密钥, 其中  $d = e^{-1} \pmod{\phi(N)}$ , 并且  $|N| = k = n + k_0 + k_1$ ,  $2^{-k_0}$  和  $2^{-k_1}$  为一个可忽略的量;  $G, H$  是两个杂凑函数, 且满足

$$G: \{0, 1\}^{k_0} \mapsto \{0, 1\}^{k-k_0}, \quad H: \{0, 1\}^{k-k_0} \mapsto \{0, 1\}^{k_0}$$

$n$  是明文消息的长度;

设  $(N, e)$  是 Alice 的 RSA 公钥,  $d$  是她的私钥。

##### 加密过程

为了发送一个消息  $m \in \{0, 1\}^n$  给 Alice, Malice 执行以下步骤计算:

1.  $r \leftarrow_U \{0, 1\}^{k_0}; s \leftarrow (m \parallel 0^{k_1}) \oplus G(r); t \leftarrow r \oplus H(s);$
2. if<sup>①</sup>  $(s \parallel t \geq N)$  go to 1;
3.  $c \leftarrow (s \parallel t)^e \pmod{N}.$

所得密文为  $c$ 。

(\* 这里, “ $\parallel$ ”表示比特串的连接, “ $\oplus$ ”表示比特按位异或 XOR 操作, “ $0^{k_1}$ ”表示  $k_1$  个 0 的串, 用做冗余, 以便解密时进行数据完整性验证。\*)

##### 解密过程

收到密文  $c$  后, Alice 执行以下步骤计算:

1.  $s \parallel t \leftarrow c^d \pmod{N}$  满足  $|s| = n + k_1 = k - k_0, |t| = k_0;$
2.  $u \leftarrow t \oplus H(s); v \rightarrow s \oplus G(u);$
3. 输出  $\begin{cases} m & \text{若 } v = m \parallel 0^{k_1} \\ \text{拒绝} & \text{其他} \end{cases}$

(\* 当输出为“拒绝”时, 就认为密文是无效的 \*)

如果一个密文具有正确的数据完整性, 那么就认为消息的发送者已经知道了所加密的明文。这个概念称为“明文感知性”。如果攻击者已经知道了所加密的明文, 那么预言机解密服务就应该不会给他提供任何新的信息, 甚至为他提供一个如何攻破目标密码系统的密码分析训练也不会给他带来任何新的信息。另一方面, 如果攻击者试图以适应性的方式修改密文, 那么数据完整性检验将以“压倒性”的概率失败, 并且解密的结果为一条无意义的消息。因此, 主动攻击者不能有效地攻击具有密文完整性保护的密码系统。

在第 14 章, 我们将介绍一个形式化模型, 用以表达在适应性选择密文攻击(CCA2)条件下的安全性概念。在第 15 章, 我们将研究一些公钥密码系统, 它们在这样的攻击下是可证明安

① 我们使用试错检验方法来确保填充结果总是小于  $N$  的整数。重复  $i$  次检验的概率为  $2^{-i}$ 。一种可选的方法是使  $r$  和  $H$ , 从而  $t$  都比  $N$  的长度少一个比特, 见 16.4.2 节中的“PSS 填充”算法。

全的, RSA-OAEP 就是其中之一。在 15.2 节中, 我们将详细分析 RSA-OAEP 加密体制的安全性。在很强的攻击条件下, 分析将是一个形式化的证明, 对适应性选择密文攻击的不可区分性, RSA-OAEP 是安全的。由于这种更强的安全性, RSA-OAEP 不再是一个教科书式的加密算法; 它是一个适于应用的公钥密码系统。

如同 RSA-OAEP 算法所说明的, 为了使密码系统能够达到 CCA2 安全, 通常的做法是为密码系统增加一个数据完整性检验机制, 而不关心消息源识别。

消息源识别是认证服务的一部分, 称为数据源认证。认证是下一章的主题。

## 10.6 本章小结

我们在本章介绍了提供数据完整性服务的基本密码技术。这些技术包括: (i) 对称技术, 它基于运用杂凑函数或分组密码算法构造的 MAC; (ii) 非对称技术, 它基于数字签名。这些技术在提供数据完整性服务的同时, 还提供了一个子服务: 消息源识别。

本章所给出的数字签名体制的安全性概念是一个教科书式版本, 因此安全性很弱。对于所介绍的一些数字签名体制, 我们也给出了早期的、有关其(教科书式的)不安全性的提醒信息。强化安全性概念和构造强的签名体制的工作将在第 16 章进行。

最后, 我们还辨别了一种特殊的数据完整性服务, 它不需要识别消息源, 并且通过一个公钥密码系统的实例来说明这种服务, 该系统运用这种服务获得了强的安全性(这里没有给出原因)。在第 15 章, 我们将看到这种特殊的数据完整性服务所起的重要作用, 利用它形成一套通用的方法, 以实现适于应用的密码系统。

## 习题

- 10.1 什么是篡改检测码(MDC)? MDC 是如何产生和怎样使用的? 消息认证码(MAC)是 MDC 吗? (消息的)数字签名是 MDC 吗?
- 10.2 什么是随机预言机? 随机预言机存在吗? 随机预言机的行为是如何逼近现实世界的?
- 10.3 设杂凑函数的输出空间大小为  $2^{160}$ , 找到该杂凑函数碰撞所的花费时间期望值是什么?
- 10.4 为什么说杂凑函数实际上是不可逆的?
- 10.5 对称和非对称数据完整性技术的主要区别是什么?
- 10.6 什么是数字签名体制的存在性伪造? 阻止存在性伪造的实用机制是什么?
- 10.7 为什么说教科书式数字签名安全性的概念是不适用的? 提示: 考虑 Rabin 签名体制在抵抗主动攻击者时的致命弱点。
- 10.8 何谓“来自 Malice 的数据完整性”安全性概念?
- 10.9 RSA-OAEP 算法(算法 10.6)的密文输出是一个有效的 MDC 吗?



# 第四部分 认 证

今天,许多商业活动、交易以及政府服务已越来越多地出现在开放且易受攻击的通信网络上,如 Internet。与期望的通信伙伴真诚地通信是至关重要的。这里需要的安全服务是认证,它可通过应用密码技术来获得。本部分分为三章,涉及各种认证协议技术。在第 11 章中,我们将研究认证协议的基本工作原理,分析认证协议中的典型错误并了解其起因。在第 12 章中,我们将分析几个重要的实用认证协议技术的个案。在第 13 章中,我们将介绍公钥基础结构的认证框架。

# 第 11 章 认证协议——原理篇

## 11.1 引言

在第 2 章中,我们已经介绍过一些认证协议,不过那里介绍的大部分协议在现实中是不存在的(只有两个协议例外):我们曾精心地设计了这些协议,使它们以不同的方式展现出各种缺陷,这是为了在密码和信息安全领域中引入一种谨慎和警觉的文化氛围。

本章,我们回到认证这一主题。再次回到该主题的目的是使我们能够对该领域进行更为广泛和深入的研究。本章内容分为两部分:

### 各种认证技术的介绍

在这一部分,我们将研究各种基本的认证技术。这些技术包含非常基本的机制和协议结构,用于消息认证和实体认证、基于口令的认证技术和一些重要的认证密钥建立技术。我们相信在国际标准中出现的一些基本的认证机制和协议结构是从大量的著作中挑选出来并且经过认真的(和长期的)专家评议过程和改进修订后形成的。因此,在介绍基本的认证技术时,我们将主要关注已被国际标准组织标准化的机制。另外,我们还将介绍其他几个著名的认证和认证密钥建立协议。我们相信在这一部分所介绍的认证机制和认证协议是有价值的,它们将会为我们以后设计好的协议提供构建模块和指南。因此,我们认为对于协议设计者而言,这一部分提供了认证技术典范。

### 认证协议缺陷的研究

这一部分是研究认证协议时必不可少的。我们将列举各种已知和典型的攻击技术,这些技术能够用于对认证协议的攻击。我们将采用可用于攻击的一些有缺陷的协议为例来讨论和分析各种攻击技术。通过这样的研究,我们就会熟悉一种常见的现象:即使由专家设计的认证协议也很可能存在安全缺陷。列举各种典型的协议缺陷和相关的攻击技术为协议设计者提供了必需的知识:“你知道这种类型的攻击吗?”

与在第 2 章中我们有意设计的、实际并不存在且有人为缺陷的虚构协议不同,本章所列举的协议中包含的安全缺陷并不是人为的,实际上,没有一个缺陷是故意设计的!这些缺陷都是在某位著名的信息安全和/或密码学专家发布了协议之后才被发现的,从而发布的协议也就成了有缺陷的协议。通过本章的学习,我们将看到这样一个事实:即使是该领域的专家,按照标准文档,遵循普遍认为很好的设计准则,甚至熟悉各种典型协议缺陷,认证协议的设计仍然是极容易出错的。

由于认证协议众所周知的易出错的本性,作为第 2 章的延续,本章和下一章依然不能成为本书中认证这一主题的终点。使用系统化方法(也就是形式化方法)来开发正确的认证协议是目前的一个重要研究课题。我们将在第 17 章中研究形式化方法,以获得正确的认证协议的一些论题。

### 11.1.1 章节概述

在 11.2 节,我们通过介绍几个细化的概念来论述认证的含义。在 11.3 节,我们将就一些惯例达成一致,包括认证协议组成部分的表示方法和协议参与者的默认行为。接下来的三节构成了本章要研究的第一部分:在 11.4 节,我们介绍用于认证协议的非常基本和标准的结构;在 11.5 节,我们学习一些基于口令的认证技术;11.6 节介绍一个重要的协议,该协议使用密码技术获得了认证和认证密钥交换这样两个目的,不过其中使用的密码技术与前面两节有所不同。我们学习的第二部分内容体现在 11.7 节,该节列出并讨论了对于认证协议典型的攻击技术。最后,在 11.8 节,我们推荐一个简短但在该领域中十分重要的参考文献列表。

## 11.2 认证和细化的概念

如果要对认证做简短的描述,我们可以说认证是个过程,通过这个过程,一个实体向另一个实体证明了某种声称的属性。例如,前者是一个主体,声称拥有某种合法权利,可以进入后者的系统或者使用后者的服务,通过认证,后者确认用户确实拥有这种权利。由这样一个简短描述,我们可以看出,认证至少涉及到两个独立的通信实体。按照惯例,协议指的就是在两方或互相协作的多方之间进行通信的过程。因此,一个认证过程也是一个认证协议。

认证这一概念可以分为三个子概念:**数据源认证**、**实体认证**和**认证的密钥建立**。第一个概念主要涉及验证消息的某个声称属性;第二个概念则更多地涉及验证消息发送者所声称的身份;第三个概念则进一步致力于产生一条安全信道,用于后继的应用层安全通信会话。

### 11.2.1 数据源认证

数据源认证(也称为**消息认证**,与数据完整性密切相关。早期的密码和信息安全教程认为这两个概念没有本质区别(例如,[90]的第 5 章,[94]的 1.2 节~1.3 节)。这种观点是基于如下考虑:使用被恶意修改过的信息和使用来源不明的消息具有相同的风险。

然而,数据源认证和数据完整性是两个差别很大的概念。这两个概念在很多方面都可以明显地区分。

首先,数据源认证必然涉及通信。它是一种安全服务,消息接收者用它来验证消息是否来源于所声称的消息源。数据完整性则不一定包含通信过程,该安全服务可以用于存储的数据。

其次,数据源认证必然涉及消息源的识别,而数据完整性则不一定涉及该过程。在 10.5 节,我们说明并论证了一个颇具说服力的例子,在无消息源识别的情况下提供数据完整性这一安全服务,我们甚至杜撰了“来自 Malice 的数据完整性”这一短语,用以标志具有这种属性的数据完整性服务。我们应该记得,根据我们在第 2 章做出的约定,Malice 是不露面的主体,其身份与某条消息的著名源信息根本不相关。在第 15 章,我们将会意识到“来源于 Malice 的数据完整性”是一种常用的机制,用于获得公钥密码体系的可证明安全。

再次,也是最重要的一点,数据源认证必然涉及确认消息的**新鲜性**,而数据完整性却无此必要:一组老的数据可能有完善的数据完整性。为了获得数据源认证服务,消息的接收者应该验证该消息是否是在新近发送的(也就是说,消息的发送和接收之间的时间间隔应该足够小)。接收者认定的在新近发送的消息通常称为新鲜的消息。要求消息的新鲜性是符合常识的:新鲜的消息意味着在通信的双方存在着一个良好的通信。并且,这一点往往进一步意味着对等



的通信方、通信设备、系统或者消息本身遭到阻挠或破坏的可能性很小。在 2.6.4 节,我们已经介绍了一个针对 Needham-Schroeder 对称密钥认证协议的攻击(Denning 和 Sacco 攻击,攻击 2.2),该攻击中的一条重放的旧消息完全具有有效的数据完整性而没有有效的认证性。这种类型的认证失败可以称为缺失消息源活现性的有效数据完整性。

一条消息是否新鲜,应该完全取决于应用。一些应用要求在一个相当短的时间间隔内才可以认为消息是新鲜的,该时间间隔通常为几秒(例如在许多基于问-答机制的实时安全通信应用中)。一些应用则在一个较长的时间段内都认为是新鲜的;例如,在第二次世界大战中,德军的军事通信使用了著名的 Enigma 密码机加密,他们规定了一个规则,即所有的密码机每天都要设置一个新的“日-密钥”[279]。这一规则已经成为当今许多安全系统中广泛应用的密钥管理准则,当然其中的“日-密钥”可能已经改变为“时-密钥”,甚至“分-密钥”。一些其他的应用对于消息的新鲜性会允许更长的时间段。例如,一张银行支票通过了完整性和源识别验证之后,它有权要求支付的有效性(真实性)取决于这张支票的使用期限,也就是从银行发行该支票之日到该支票过期之时的时间间隔。大部分银行允许支票有三个月的使用期限。

最后,我们指出由一些密码体制(如盲签名)支持的匿名证件也为区分数据源认证和数据完整性提供了很好的例子。匿名证件可以发行给用户,这使得持有者能够匿名地向系统证明其成员身份,从而获得该系统的某种服务。这里应该注意到,数据完整性的证据甚至能够以真实通信的方式证明,然而,该系统是不允许执行源识别的。我们在稍后的章节中会研究这种密码技术。

根据迄今为止的讨论,我们可以将数据源认证这一概念的特征总结如下:

- i) 包含从某个声称的源(发送者)到接收者的消息传输过程,该接收者在接收时会验证消息。
- ii) 接收方执行消息验证的目的在于确认消息发送者的身份。
- iii) 接收方执行消息验证的目的还在于确认在原消息离开消息发送者之后的数据完整性。
- iv) 验证的进一步目的在于确认消息传输的“活现性”。

### 11.2.2 实体认证

实体认证是一个通信过程(如协议),通过这个过程某个实体和另一个实体建立一种真实通信,并且第二主体声称的身份应该和第一主体所寻求的通信方一致。通常,“实体”这个词可以忽略,像下面的陈述中:“认证协议的一个重要目标是确认某个主体的真实通信。”

通常,声称的身份信息在协议中也完全能单独构成消息。这种情况下,关于声称身份的活现性可以通过应用数据源认证机制来确认。实际上,正如我们将会在本章的许多场合看到的,声称的身份信息在协议中单独构成消息时,把该消息作为数据源认证来处理,确实是构成获得实体认证的一种合适方法。

根据协议主体的不同归类方法,在分布式系统中,实体认证可以分为若干类型。这里我们列出常用的几类,当然并没有穷尽所有可能的情况。

**主机-主机类型** 通信的参与者是在分布式系统中被称为“节点”的计算机或者平台。主机级别的活动经常需要主机间的协作。例如,在远程“重启”<sup>①</sup>的平台中,在重启时,该平台必

---

① “重启”是计算机科学中技术上的术语,它是指重新初始化计算机系统,这种初始化是从一些简单的指令或者系统中某些固化的信息集开始的。

须能够识别可信服务器来提供必需的信息,例如操作系统的可信拷贝、可信的时钟设置或者当前可信的环境设置。可信信息的确认通常通过运行认证协议来完成。这种主机-主机类型通信的通常实例是**客户端-服务器**设置,其中一台主机(客户端)向另一台主机(服务器)请求某些服务。

**用户-主机类型** 用户通过登录系统中某台主机来获得访问该计算机系统。最简单的例子是通过 telnet 登录到某台计算机或者通过 ftp(文件传输协议)执行文件传送;这两种情况都能通过运行某个口令认证协议来完成。而在更为严格的应用中,因为不安全的主机会导致严重的损失(例如当用户通过智能卡进行电子支付时),**双方认证**就是必需的。

**进程-主机类型** 现在,分布式计算已经取得了长足的进展,这使得大量的功能和服务成为可能。一个主机可能会给外部的进程授予不同的接入权限。例如,某段“移动代码”或者“JAVA™程序”<sup>①</sup>都能够到达远程主机并作为远程进程在该主机上运行。在敏感的应用中,设计认证机制是必需的,也是可能的,这样才能使主机识别外部进程是否友好,从而能够对外部进程赋予合适的接入权限。

**成员-俱乐部类型** 可以把成员拥有俱乐部证书的证明看做是一般化的“用户-主机类型”。这里,俱乐部可以只需要考虑成员证件的有效性,而没有必要知道该成员的进一步信息,如该成员的真实身份。运用零知识识别协议和不可否认签名方案可以实现这类实体认证。我们将在第 18 章学习这些认证技术。

### 11.2.3 认证的密钥建立

通常,通信方运行实体认证协议的目的在于能够在高层或者应用层上进行安全通信。在现代密码学中,密钥是安全通信信道的基础。因此,为了进行高层或应用层安全通信而运行的实体认证协议通常都有一个子任务,即(认证的)密钥建立,或者**密钥交换**、**密钥协商**。

如实体认证的情况,可根据数据源认证获得有关宣称者的身份,在认证的密钥建立协议中,密钥建立素材也是重要的协议消息,因而它应该也是数据源认证的内容。

在文献中,(实体)认证协议、认证的密钥建立(密钥交换,密钥协商)协议、安全协议或有时甚至密码协议,常常都是指同一通信协议集。

### 11.2.4 对认证协议的攻击

因为认证协议(包括数据源认证、实体认证、认证的密钥建立)的目的在于证明某种声称的属性,因而就不可避免要用到密码技术。同样不可避免的是,认证协议的目的总是和其对立物(攻击)相伴而生。对认证协议的攻击包括未经授权而企图获益的攻击者或共谋者(这些主体我们通称为 Malice,见 2.3 节)。这种获益可能造成严重的后果,例如 Malice 获得机密信息或者密钥,也可能只造成较小的损害,例如 Malice 成功地欺骗某个主体对某个宣称属性做出错误判断。通常,如果某个主体断定自己和意定的通信方正常运行了协议,而意定的通信方却有不同的结论,那么就认为该协议存在缺陷。

我们必须强调,对于认证协议的攻击主要是指那些不涉及破解底层密码算法的攻击。通常,认证协议不安全不是因为该协议所用的底层密码算法很弱,而是因为协议设计上的缺陷,

<sup>①</sup> JAVA™程序是一段可执行的代码,该代码在远程主机上由“Web 浏览器”运行,为的是代表代码发行方主机实现某种功能。

这些缺陷使得 Malice 能够在不需要破解密码算法的条件下破坏认证的目的。本章我们将会看到许多这样的攻击实例。因此,在分析认证协议时,我们通常假设底层的密码算法是“完善的”,不考虑其可能存在的弱点。这些弱点通常在密码学的其他论题中考虑。

### 11.3 约定

在本章其余部分出现的认证协议中,我们将规定一些约定,用于确定协议消息的语法结构所表达的语义。约定列举如下:

- $Alice, Bob, Trent, Malice, \dots$ : 在协议消息中出现的主体名称。有时可以简写为  $A, B, T, M, \dots$ ;
- $Alice \rightarrow Bob: M$ ; Alice 给 Bob 发送消息  $M$ ; 协议规范就是几个这种消息通信的序列;
- $\{M\}_K$ : 用密钥  $K$  加密消息  $M$  得到的密文;
- $K, K_{AB}, K_{AT}, K_A, \dots$ : 密码密钥, 其中  $K_{XY}$  表示主体  $X$  和  $Y$  共享的密钥,  $K_X$  表示主体  $X$  的公钥;
- $N, N_A, \dots$ : 随机数, 表示“使用一次的数”[62]; 这些随机数是从一个足够大的空间中抽样得到的。  $N_X$  表示该随机数由主体  $X$  生成;
- $T_X$ : 主体  $X$  创建的时戳;
- $Sig_A(M)$ : 主体  $A$  所创建的对消息  $M$  的数字签名。

**注释 11.1** 我们应该注意上面阐述的协议消息的语义, 该语义是与其语法结构(类型)相联系的, 然而该语义对于协议的参与主体(例如 Alice)来说并非必须是可理解的。一般来说, 对于某个协议中的某条任意消息或者该消息的一部分, 如果协议规定并没有要求 Alice 对该消息或者这部分消息执行密码操作, 那么 Alice(实际上是她的协议编译器)对于这部分消息的理解只限于语法结构层次。在语法层次, Alice 很可能对协议消息的语义做出错误的解释。在例 11.1 中, 我们会举例说明各种可能的错误理解。 □

**例 11.1** 在语法层次, Alice 可能会对协议的消息做出错误的解释。举例如下:

- 她可能会把某部分消息误认为密文, 并且如果她认为自己拥有正确的密钥, 她可能会试图对这部分消息解密, 或者如果她认为这部分消息应该给 Bob, 她会把它转发给 Bob。然而, 这部分消息可能仅仅是某个主体的身份(例如 Alice 或 Bob), 或者是一个随机数, 或者是时戳。
- 她可能会“按照协议的规定”把密文进行解密并把结果发送出去, 然而被解密的密文可能是她以前在某个不同的环境下创建的。
- 她可能会把某个密钥参数误认为某个随机数; 等等。

这么看起来 Alice 好像在理解协议消息方面很“愚笨”。其实不然, 我们应该认为 Alice 是天真的, 因为她不是总能预测到存在一个“聪明”的 Malice, 他已经通过把协议消息的各个部分按照错误的顺序“重新编排”过了, 以至于造成了错误的解释。 □

通常, 对于协议参与者的主体行为, 包括合法的主体和不合法的主体, 我们有一组进一步的约定:

- 协议中的诚实主体在协议成功终止之前不能理解任何协议消息的语义。
- 协议中的诚实主体不能识别、创建或者分解  $\{M\}_K$ , 除非该主体拥有正确的密钥。
- 协议中的诚实主体不能识别看似随机的数据, 比如随机数、序列号或者密钥, 除非随机化的数据是在当前协议的运行中由该主体创建的, 或者是该协议运行完成后给该主体的某个输出。
- 协议的诚实主体不主动记录任何协议消息, 除非协议规定该主体必须记录。通常认证协议是无状态的, 也就是说, 在协议运行成功结束之后, 该协议不需要参与主体维护任何状态信息, 除非该信息是协议运行后给参与主体的某个输出。
- 除了在 2.3 节中规定的能力外, Malice 还知道诚实主体的“愚笨”(公平地说, 是弱点), 我们在例 11.1 中已经就这种“愚笨”做了说明, 并且 Malice 总是想利用这些弱点。

认证协议意味着要在公开通信网络中传输消息, 并假设该通信网络在 Malice 的控制之下。认证协议还意味着要能防止这种环境下的攻击, 尽管此时 Malice 是“聪明的”, 而诚实主体是“愚笨的”。

接下来我们来看认证协议是如何做到这一点的。

## 11.4 基本认证技术

现实中有许多基于协议的技术用于实现认证(数据源认证、实体认证)和认证的密钥建立。然而, 基本的认证结构, 尤其是被认为很好的结构, 和这些好的认证结构所包含的简单技术思路, 却不是很多。

本节我们通过介绍一些基本的、重要的协议结构来研究基本的认证技术。在研究过程中, 我们将主要关注在一系列的国际标准中已证明过的结构。我们认为这些结构可以作为设计认证协议的模板。我们也将论证为什么某些结构比其他结构更合适, 并用例子来给出一些不好的结构, 解释其不好的理由。

本节将研究以下基本的认证技术:

- 证明消息的新鲜性和主体活现性的标准机制(11.4.1 节)
- 双方认证和单方认证的比较(11.4.2 节)
- 包含可信第三方的认证(11.4.3 节)

### 11.4.1 消息新鲜性和主体活现性

判断某条消息是否新鲜是数据源认证必不可少的一部分(这里请注意在 11.2.1 节中讨论过的消息源识别和数据源认证的区别), 同样在实体认证中, 主体也要考虑与意定通信方通信的真实性。因此, 证明消息新鲜性或主体活现性的机制就成为认证协议中最基本的一个组成部分。

下面我们描述实现这些功能的基本和标准的机制。在描述中, 我们假设 Alice 是关于某种属性的声称者(例如, 她的活现性, 或者消息的新鲜性); Bob 是验证者, 他验证某种声称的属性。如果机制中使用的是对称密码技术, 我们就假设 Alice 和 Bob 共享某个密钥  $K_{AB}$ ; 如果机制用的是非对称密码技术, 我们就假设 Bob 通过公钥证书框架<sup>①</sup>知道了 Alice 的公钥。

<sup>①</sup> 公钥证书框架将在第 13 章介绍。

### 11.4.1.1 询问-应答机制

在询问-应答机制中, Bob(验证者)在协议消息的组合中拥有他的输入消息, 并且该合成消息涉及了 Alice(声称者)所进行的密码操作, 于是 Bob 能够通过他自己输入的消息的新鲜性来验证 Alice 通信的真实性。Bob 输入的通常形式可以是他生成的某个随机数(称为一次性随机数)并且预先传送给 Alice。假设  $N_B$  代表 Bob 生成的一次性随机数。这种消息新鲜性机制通常具有以下交互形式:

1. Bob  $\rightarrow$  Alice:  $N_B$ ;
  2. Alice  $\rightarrow$  Bob:  $\mathcal{E}_{K_{AB}}(M, N_B)$ ;
  3. Bob 解密接收到的密文分组并  $\begin{cases} \text{接受} & \text{如果 Bob 看到 } N_B \\ \text{拒绝} & \text{其他情况。} \end{cases}$
- (11.4.1)

这里, 发送的第一条消息通常称为 Bob 对 Alice 的询问, 而第二条发送的消息也因此称为 Alice 对 Bob 的应答。Bob 是发起者而 Alice 是响应者。

以上给出的机制使用了对称密码技术: 对称加密。因此, 接收到 Alice 的应答以后, Bob 必须使用共享密钥  $K_{AB}$  来解密收到的密文分组。如果解密后正确提取出了 Bob 的一次性随机数(一定要注意“正确”的含义, 稍后我们会看到, 它实际上意味着正确的数据完整性), 那么 Bob 就能够断定 Alice 确实在他发出询问这一动作以后执行了所要求的密码操作; 如果询问和应答之间的时间间隔(由应用的需求决定, 我们在 11.2.1 节已经讨论)是可以接受的, 就认为该消息  $M$  是新鲜的。这种消息新鲜性机制的直观基础是确信 Alice 的密码操作肯定是在接收到 Bob 的一次性随机数之后进行的。这是因为 Bob 的一次性随机数是在一个足够大的空间中抽样得到的, 所以没有人能够在抽样之前预测该值。

现在让我们解释 Bob“正确地”(在前面段落中我们已经给出了警告)解密和提取他的一次性随机数的含义。该机制中使用的对称加密可能会使人们觉得这里提供的安全服务是机密性。然而事实上, 实现消息新鲜性时所必需的安全服务应是数据完整性。读者可能要争论, 两个主体可能希望保持  $M$  的机密性, 例如,  $M$  可能是以后确保高层通信会话安全的密码密钥(因此这个基本结构包含有会话密钥建立的子任务)。这确实是采用加密技术的一个合理理由。我们实际上能够进一步认为通信双方可能也要保持 Bob 的一次性随机数的秘密性, 因而 Bob 也应该加密发送的第一条消息。因此, 我们在这里不是说当需要加密服务时使用加密提供机密性是错的。这里所强调的只是如果加密算法没有提供合适的数据完整性服务(加密算法通常是不会提供的), 那么应用上面的机制就会因为缺少必要的数据完整性服务而面临危险! 在 17.2.1 节中, 我们会有令人信服的证据看出下述注释背后的缘由:

**注释 11.2** 如果认证机制(11.4.1)中的加密算法没有提供合适的数据完整性服务, 那么 Bob 就不能确认消息  $M$  的新鲜性。 □

使用对称密码技术实现数据完整性服务的真正正确和标准的方法就是使用篡改检测码(MDC, 见 10.1 节的定义 10.1)。因此, 在机制(11.4.1)中加密应附加上 MDC, 其中 MDC 由共享密钥控制, 输入是共享密钥和需要进行完整性保护的密文分组。如果消息  $M$  不需要机密性保护, 那么下面的机制就是实现消息新鲜性的一种合适机制:



1. Bob  $\rightarrow$  Alice:  $N_B$ ;
2. Alice  $\rightarrow$  Bob:  $M, \text{MDC}(K_{AB}, M, N_B)$ ;
3. Bob 重构  $\text{MDC}(K_{AB}, M, N_B)$  并  $\begin{cases} \text{接受} & \text{如果两个 MDC 相同} \\ \text{拒绝} & \text{其他情况} \end{cases}$

(11.4.2)

注意,为了 Bob 能够在第 3 步重构 MDC,现在消息  $M$  在第 2 步中必须以明文发送。当然  $M$  本身可以是某个机密消息的密文。

在 17.2.1 节,我们会以令人信服的证据论证在使用对称密码技术实现认证时,机制(11.4.2)是正确的方法而机制(11.4.1)是不正确的。在那里我们也会看到,即使(11.4.1)中的机制使用了强安全的加密算法,如果没有合适的数据完整性, $M$  的机密性也不一定能得到保证。

询问-应答机制也可以采用非对称密码技术来实现,该机制如下:

1. Bob  $\rightarrow$  Alice:  $N_B$ ;
2. Alice  $\rightarrow$  Bob:  $\text{sig}_A(M, N_B)$ ;
3. Bob 使用他的一次性随机数验证签名并  $\begin{cases} \text{接受} & \text{如果通过了签名验证} \\ \text{拒绝} & \text{其他情况} \end{cases}$

(11.4.3)

注意在该机制中, Alice 能够自由地选择  $M$  是很重要的。Alice 对  $M$  的自由选择是防范措施的一部分,用于防止 Bob 利用该机制来欺骗 Alice 使她不经意地为 Bob 准备好消息签名。例如, Bob 可能已经按照下面的方式准备好了他的“一次性随机数”:

$$N_B = h(\text{从 Alice 号码为 456 的账户取 1000 美元存入 Bob 号码为 123 的账户})$$

其中  $h$  是某个杂凑函数。

在某些应用中,执行机制(11.4.3)的 Alice 作为签名者可能不能自由地选择  $M$ 。此时,应该定义专门的密钥以对密钥的使用做出限定。例如,在机制(11.4.3)中,对于用来验证 Alice 的签名的公钥,应该指明它的这一特殊用途。对密钥的使用做出说明是实际中**密钥管理**的内容。

#### 11.4.1.2 询问-应答机制标准化

ISO(国际标准化组织)和 IEC(国际电子协会)已经把我们目前所阐述的三种询问-应答机制标准化为**单方实体认证机制**的基本结构。机制(11.4.1)标准化后称为“ISO 两次传输单方认证协议”,如下所示[149]:

1.  $B \rightarrow A: R_B \parallel \text{Text1}$ ;
2.  $A \rightarrow B: \text{TokenAB}$ 。

这里  $\text{TokenAB} = \text{Text3} \parallel \mathcal{E}_{K_{AB}}(R_B \parallel B \parallel \text{Text2})$ 。

在接收到  $\text{TokenAB}$  后, Bob 应该对它解密;如果解密后正确地显示了 Bob 的一次性随机数  $R_B$ ,他就应该接受这次运行,否则拒绝这次运行。

在这里和以下的 ISO/IEC 标准中,我们将准确地使用 ISO/IEC 用于规范协议的标准符号。在 ISO/IEC 规范中,  $\text{Text1}$ 、 $\text{Text2}$  等属于可选项,  $\parallel$  表示比特链接,  $R_B$  表示 Bob 生成的一次性随机数。

这里我们请读者再次注意到以下事实的重要性:为加密算法提供数据完整性服务是一个必需的条件,用于检测解密结果是否正确(回顾 11.4.1.1 节的注释 11.2)。



还应该注意到的是,我们认为(11.4.1)是基本的消息新鲜性机制,然而在 ISO/IEC 标准中对应的却是实体认证机制。因此包含消息“B”,也就是 Bob 的身份,用于代替(11.4.1)中的  $M$  是至关重要的:这种包含使以下事实明显成立,即 ISO/IEC 机制的目的是证明 Bob 的真实通信,是实体认证协议,其中 Bob 是认证的主体。Abadi 和 Needham 在[1]中列举了有关密码协议设计的一些谨慎的工程准则:明确意定的认证主体的身份是其中一条重要的准则。在 11.7.7 节中,我们会看到在认证协议中忽略该主体的身份所带来的危险。

机制(11.4.2)的 ISO/IEC 标准化版本称为“使用密码验证函数(CCF)的 ISO 两次传输单方认证协议”。如下所示[151]:

1.  $B \rightarrow A: R_B \parallel \text{Text1};$
2.  $A \rightarrow B: \text{TokenAB}.$

这里<sup>①</sup> $\text{TokenAB} = \text{Text2} \parallel f_{K_{AB}}(R_B \parallel B \parallel \text{Text2}); f$  是某种 CCF,本质上是某种密码学杂凑函数。这里使用的 CCF 是带有密钥的。

接收到  $\text{TokenAB}$  之后,  $B$  应该使用共享密钥、他产生的一次性随机数、他的身份信息和  $\text{Text2}$  重新构造带密钥的 CCF,如果  $B$  重构的 CCF 分组和接收的分组相同,  $B$  就应该接受这次运行,否则拒绝这次运行。

机制(11.4.3)的 ISO/IEC 标准化版本称为“使用公钥的 ISO 两次传输单方认证协议”,如下所示[150]:

1.  $B \rightarrow A: R_B \parallel \text{Text1};$
2.  $A \rightarrow B: \text{CertA} \parallel \text{TokenAB}.$

这里  $\text{TokenAB} = R_A \parallel R_B \parallel B \parallel \text{Text3} \parallel \text{sig}_A(R_A \parallel R_B \parallel B \parallel \text{Text2}); \text{CertA}$  是 Alice 的公钥证书(我们将在后面的章节研究公钥证书)。

在接收到  $\text{TokenAB}$  之后,  $B$  应该验证签名;如果签名通过验证,  $B$  应该接受这次运行;否则拒绝这次运行。

和我们前面关于机制(11.4.3)的讨论一样,在 ISO/IEC 协议中,  $A$  能够自由地选择  $R_A$  构成了防范措施的一部分,用于防止  $A$  不经意地对 Bob 已准备好的消息签字。

### 11.4.1.3 时戳机制

在时戳机制中, Alice 把当前时间加入到合成消息中,该合成消息会涉及某种密码操作,这样,当前时间就通过密码操作综合到了她的消息中。

假设  $T_A$  表示 Alice 创建的时戳, Alice 在合成她的消息时会创建该时戳。这种消息新鲜性机制具有如下的非交互形式:

1.  $\text{Alice} \rightarrow \text{Bob}: \mathcal{E}_{K_{AB}}(M, T_A);$
2. Bob 解密密文分组并  $\begin{cases} \text{接受} & \text{如果 } T_A \text{ 认为是有效的} \\ \text{拒绝} & \text{其他情况} \end{cases} \quad (11.4.4)$

<sup>①</sup> 在[151]中,明文部分的  $\text{Text2}$  被错误地写成了  $\text{Text3}$ 。如果没有明文形式的  $\text{Text2}$ ,  $B$  就不能通过重构来验证 CCF。

与机制(11.4.1)类似,Bob 进行的解密操作必须检验数据完整性的正确性(回顾 11.4.1.1 节和那里的注释 11.2)。在解密后,Bob 把得到的  $T_A$  和本地时间进行比较(我们假设该协议的主体使用全局标准时间,比如格林威治平均时间)。如果 Bob 认为比较结果所显示的时差足够小,那么就认为  $M$  是新鲜的。

类似 11.4.1.1 节中我们的评述,没有数据完整性保护的加密是安全服务的误用。因此,使用对称加密技术的时戳机制应该有下面更为合理的版本:

1. Alice  $\rightarrow$  Bob:  $M, T_A, \text{MDC}(K_{AB}, M, T_A)$ ;
2. Bob 重构  $\text{MDC}(K_{AB}, M, T_A)$  并
 

$\left\{ \begin{array}{ll} \text{接受} & \text{如果两个 MDC 是相等的并且 } T_A \text{ 是有效的} \\ \text{拒绝} & \text{其他情况} \end{array} \right.$

(11.4.5)

在这个版本中,Bob 通过检查对时戳和消息进行密码综合的单向变换方式,来完成数据完整性验证。当然,如果  $M$  还需要机密性保护,那就必须使用加密。然而使用加密并不能排除使用数据完整性校验的必要性。

显然,使用非对称密码技术也能够实现时戳机制。

1. Alice  $\rightarrow$  Bob:  $\text{sig}_A(M, T_A)$ ;
2. Bob 验证签名并
 

$\left\{ \begin{array}{ll} \text{接受} & \text{如果签名通过验证并且 } T_A \text{ 是有效的} \\ \text{拒绝} & \text{其他情况} \end{array} \right.$

(11.4.6)

时戳机制避免了交互的需求,因而适用于不涉及交互的应用,例如在电子邮件中的应用。然而,时戳机制的缺点在于需要同步时钟,并且必须对时钟进行安全维护,这可能会很困难。关于时戳的困难、警告和反对意见在文档[29,35,117,100]中有很好的论述。

到目前为止,在我们所介绍的协议的基本结构中,一次性随机数或者时戳都是消息的特殊组成部分。它们所起的作用在于识别其他消息的新鲜性,这些消息以密码的方式同随机数或时戳综合在一起。以后我们就用新鲜性标志符来表示一次性随机数或者时戳。

#### 11.4.1.4 时戳机制的标准化

ISO/IEC 对用于认证协议的时戳机制也进行了标准化。

机制(11.4.4)的 ISO/IEC 标准称为“ISO 对称密钥一次传输单方认证协议”[149],如下所示:

1.  $A \rightarrow B$ :  $\text{Token}_{AB}$ 。

这里  $\text{Token}_{AB} = \text{Text2} \parallel \mathcal{E}_{K_{AB}} \left( \begin{array}{c} T_A \\ N_A \end{array} \parallel B \parallel \text{Text1} \right)$ 。

再一次地,因为这个简单的机制使用了加密-解密的方法,我们应该回顾 11.4.1.1 节中的注释 11.2 关于加密算法对于数据完整性保护的重要性。

其中, $T_A$  表示可以选择时戳  $T_A$  或者序列号  $N_A$ 。在使用序列号时,Alice 和 Bob 维护某个同步的序列号,这样,序列号  $N_A$  应该以一种 Bob 知道的方式增加。在对序列号成功地接收和验证之后,两个主体中的每一方都应该将序列号管理器更新到某个新的状态。

序列号机制有两个缺点。首先,对于每一个可能存在的通信方都必须维护一系列的状态信息。这在开放环境下的应用中可能会比较困难,这种环境下一个主体可能会和其他多个主

体进行通信。因此,序列号机制在规模上大不了。其次,管理序列号在通信出错的时候会很麻烦,这种错误可能真的是通信传输的错误,也可能是人为的(例如,拒绝服务攻击造成的)。注意到我们在 11.3 节中约定认证协议应该是无状态的;有状态的协议在恶意环境中不能很好地工作。因此我们不主张使用序列号机制,虽然这样的机制已写入 ISO/IEC 标准文件中。

机制(11.4.5)的 ISO/IEC 标准称为“使用密码验证函数的 ISO 一次传输单方认证”[151],如下所示:

1.  $A \rightarrow B: \text{Token}AB$ 。

这里<sup>①</sup> $\text{Token}AB = \frac{T_A}{N_A} \parallel B \parallel \text{Text1} \parallel f_{K_{AB}}(\frac{T_A}{N_A} \parallel B \parallel \text{Text1})$ ;  $f$  是带有密钥的 CCF,例如,带有密钥的杂凑函数。

读者可能已经能够猜到下面命名的协议与加密和密码验证函数的公钥形式相对应,称为“ISO 公钥一次传输单方认证协议”[150]:

1.  $A \rightarrow B: \text{Cert}A \parallel \text{Token}AB$ 。

这里  $\text{Token}AB = \frac{T_A}{N_A} \parallel B \parallel \text{Text2} \parallel \text{sig}_A(\frac{T_A}{N_A} \parallel B \parallel \text{Text1})$ 。

#### 11.4.1.5 非标准机制

到目前为止,我们已经介绍了用于构建认证协议的几种基本结构。不难想像,还存在很多其他类型的结构,这些结构同样也能实现前面介绍的基本结构所实现的目的。例如,机制(11.4.1)使用对称密码技术的一种变形可以是:

1.  $\text{Bob} \rightarrow \text{Alice}: \text{Bob}, \mathcal{E}_{K_{AB}}(M, N_B)$ ;
2.  $\text{Alice} \rightarrow \text{Bob}: N_B$ ;
3.  $\text{Bob} \begin{cases} \text{接受} & \text{如果返回的一次性随机数是正确的} \\ \text{拒绝} & \text{其他情况} \end{cases}$

(11.4.7)

另一个例子,机制(11.4.3)使用的是非对称密码体制的一种变形可以是:

1.  $\text{Bob} \rightarrow \text{Alice}: \mathcal{E}_{K_A}(M, \text{Bob}, N_B)$ ;
2.  $\text{Alice} \rightarrow \text{Bob}: N_B$ ;
3.  $\text{Bob} \begin{cases} \text{接受} & \text{如果返回的一次性随机数是正确的} \\ \text{拒绝} & \text{其他情况} \end{cases}$

(11.4.8)

这里  $\mathcal{E}_{K_A}$  表示使用 Alice 公钥的公钥加密算法。在这两个变形中, Bob 验证 Alice 的真实通信的方法是加密一个新鲜性标志符,然后检验 Alice 能不能执行适时的解密。我们以后用加密-解密(新鲜性标志符)来表示这种机制。

虽然执行新鲜性标志符的加密-解密确实提供了一种验证方法,用于验证某个意定通信方的真实通信,然而这样的机制在构造认证协议时并不是令人满意的。在这种机制中, Alice 能够被用做一个解密预言机(预言机服务的含义见 7.8.2.1 节和 8.9 节),因而会不经意地泄漏

<sup>①</sup> 和前一脚注一样,在[151]中也错误地把明文部分的 Text1 写成了 Text2,这样会使得 B 不能验证 CCF。

机密信息。例如, Malice 可能记录了来源于 Alice 和 Bob 某次机密通信的某个密文分组, 然后把这个分组嵌入到某个协议中(该协议使用加密-解密机制); 接下来, Alice 就可能会被欺骗, 进而泄漏机密的会话内容。回忆我们对于诚实主体的约定(在 11.3 节): Alice 可能会错误地把某条消息认为是一次性随机数, 并忠实地按照“协议的指示”, 返回该“一次性随机数”。

加密-解密机制的这种不好的特性也在下面的事实中反映出来, 那就是 ISO/IEC 标准化时没有考虑标准化这样一种机制。这也是我们把机制(11.4.7)和(11.4.8)称为非标准机制的部分原因。

然而, 许多认证协议确实是设计使用了某种加密-解密机制。在 17.2 节, 我们会分析几个这样的协议。在那里, 我们会指出那些协议中存在的安全缺陷主要是因为使用了非标准的机制。

### 11.4.2 双方认证

到目前为止, 介绍的用于消息新鲜性或者主体活现性的基本机制实现的都是所谓的“单方认证”, 意思是说协议的两个参与主体中只对其中的一个主体进行认证。在双方认证中, 两个通信实体要互相认证。

ISO 和 IEC 已经标准化了许多用于双方认证的机制。在协议 11.1 中, 我们给出了一种基于签名的机制, 称为“ISO 公钥三次传输双方认证协议”[150]。我们给出该机制是为了澄清对于双方认证一个普遍误解。

#### 协议 11.1 ISO 公钥三次传输双方认证协议

前提: A 拥有公钥证书  $Cert_A$ ;

B 拥有公钥证书  $Cert_B$ ;

目标: A 和 B 完成双方认证。

1.  $B \rightarrow A: R_B$ ;
2.  $A \rightarrow B: Cert_A, TokenAB$ ;
3.  $B \rightarrow A: Cert_B, TokenBA$ 。

这里

$$TokenAB = R_A \parallel R_B \parallel B \parallel \text{sig}_A(R_A \parallel R_B \parallel B);$$

$$TokenBA = R_B \parallel R_A \parallel A \parallel \text{sig}_B(R_B \parallel R_A \parallel A)。$$

(\* 省略可选择的文本部分 \*)

人们可能会认为双方认证就是简单的两次单方认证; 换句话说就是, 双方认证能够通过两次应用 11.4.1 节中某个单方认证协议来实现, 一个方向一次。然而, 这样做一般并不可靠。

在 ISO/IEC 对于协议 11.1 标准化的早期, 对于双方认证和单方认证的微妙关系并没有清楚地理解。在协议 11.1[145, 132]的几个早期的标准化草案中,  $TokenBA$  和当前的版本有轻微的不同:

$$TokenBA = R'_B \parallel R_A \parallel A \parallel \text{sig}_B(R'_B \parallel R_A \parallel A)$$

在早期的草案中, 特意不允许 B 重复使用他用来提问的一次性随机数  $R_B$ , 为的是防止 B 会对一个预先已经被部分定义并完全被 A 所知的字符串签名。除这种合理的考虑外, 早期版本中

的  $\text{Token}_{AB}$  和  $\text{Token}_{BA}$  在语法上是完全对称的,这种情况在 ISO/IEC 9798-3 的几个版本中一直存在,直到 ISO 中的加拿大成员 Wiener 发现了对这种情况的一种攻击 [145] (见 [200] 中的 12.9 节)。该攻击因此被称为“加拿大人攻击”。除 ISO 标准文档外,Diffie、van Oorschot 和 Wiener 在 [100] 中讨论了该攻击。因此我们也把该攻击称为 Wiener 攻击。

#### 11.4.2.1 Wiener 攻击(加拿大人攻击)

Wiener 对于“ISO 公钥三次传输双方认证协议”早期版本的攻击如攻击 11.1 所示(回忆我们在 2.6.2 节所约定的符号,其中描述了 Malice 以伪装的方式发送和截取消息)。

**攻击 11.1** Wiener 对于 ISO 公钥三次传输双方认证协议的攻击

前提:除了协议 11.1 中的前提外, Malice 也拥有证书  $\text{Cert}_M$ ;

1.  $\text{Malice}("B") \rightarrow A: R_B$
2.  $A \rightarrow \text{Malice}("B"): \text{Cert}_A, R_A \parallel R_B \parallel B \parallel \text{sig}_A(R_A \parallel R_B \parallel B)$
- 1'.  $\text{Malice}("A") \rightarrow B: R_A$
- 2'.  $B \rightarrow \text{Malice}("A"): \text{Cert}_B, R'_B \parallel R_A \parallel A \parallel \text{sig}_B(R'_B \parallel R_A \parallel A)$
3.  $\text{Malice}("B") \rightarrow A: \text{Cert}_B, R'_B \parallel R_A \parallel A \parallel \text{sig}_B(R'_B \parallel R_A \parallel A)$

后果:

A 认为是 B 发起了这次运行,并接受了 B 的身份;而 B 实际上并没有发起这次运行,并且在等待结束由 Malice("A")发起的运行。

在发现 Wiener 攻击以后,ISO/IEC 9798 系列用于标准化认证协议时对双方认证开始采取了谨慎的方法。如果  $\text{Token}_{AB}$  在某个单方认证协议中出现,而双方认证又是在该单方认证版本的基础上发展的,那么在该双方认证中  $\text{Token}_{AB}$  的匹配  $\text{Token}_{BA}$  用于双方认证时会包含一个到  $\text{Token}_{AB}$  的上下文相关链接,该链接通常是通过重用相同的(也就是,当前的)运行中的新鲜性标志符来实现。

在当前的“ISO 公钥三次传输双方认证协议”(即协议 11.1,其中已经对容易受到 Wiener 攻击的早期版本做了修正)中,明确要求 A 在当前协议运行终止之前,要维护关于 B 的一次性随机数  $R_B$  的状态信息。

#### 11.4.3 包含可信第三方的认证

到目前为止,我们在本章所介绍的认证协议的基本结构中,均假设协议的参与双方要么已经预先共享某条安全信道(在使用对称密码技术的结构中),要么一方预先知道另一方的公钥(在使用非对称密码技术的结构中)。所以我们可以说那些基本结构都仅适合于预先互相认识的参与主体。既然如此,他们又为什么要运行认证协议呢?一个简单的答案是,他们希望通过重新确认他们之间的真实通信来更新安全信道。

另一个较好的答案是,这些基本协议结构实际上是认证协议的组成模块,这些认证协议适于开放环境中更一般、更标准的通信模式。

开放系统中的标准通信模式是指参与主体的“交互-遗忘”模式。因为开放系统很大,所以要求某个开放系统中的主体维护它和系统中所有其他主体的通信状态是很困难的。如果两个互不相识的主体希望进行安全通信,他们应该首先建立一个安全信道。在现代密码学中,安全通信信道是由密钥支撑的。因此,希望在他们之间建立安全信道的两个主体应该运行某种认证协议,这种认证协议具有建立认证的密钥的子任务。这样的协议称为认证的密钥建立协议。在完成由认证密钥支撑的安全通信会话之后,通信双方应该立即删除所建立的信道。这里“删除所建立的信道”是指参与主体把支撑安全信道的密钥遗忘,并且永不重用该密钥。这就是我们通常把一个认证的密钥建立协议输出的安全信道称为会话信道,而把支撑该信道的密钥称为会话密钥的原因。

主体在开放环境中运行认证和密钥建立协议的标准架构是使用来自于可信第三方(TTP)的集中化认证服务。这种 TTP 服务可以是在线的,也可以是离线的。下一章我们将介绍使用离线 TTP 认证服务的认证框架。

认证服务由在线 TTP 提供时,该 TTP 会与系统中或子系统的大量用户有长期的联系。在线 TTP 架构下,认证和/或认证的密钥建立协议是在 11.4.1 节和 11.4.2 节中介绍的基本结构的基础上构建的,其中“已经互相认识”的两个主体中的一个 TTP,另一个是 TTP 的某个用户。TTP 执行的密码操作可能暗示或者引导它的某个用户进行某种正确的密码操作。在 TTP 的帮助下,任意两个用户之间,即使完全不相识,也能够建立安全通信。在第 2 章我们已经看到了几个类似协议,在那里我们称 TTP 为 Trent。

ISO/IEC 的标准化认证协议(9798 系列)有两个标准的结构需要在线的可信第三方的参与[149]。其中的一个称为“ISO 四次传输认证协议”,另一个称为“ISO 五次传输认证协议”。这两个协议都实现了双方实体认证和认证的密钥建立。然而,鉴于以下两个原因,这里我们不再给出这两个协议。

第一,这些协议建立在我们于 11.4.1 节和 11.4.2 节中介绍的基本结构的基础上,因而在提供设计准则方面,这些协议不能给我们提供什么新的思想,并且在引导我们深入研究这一主题方面也没有什么积极效果。与此相反,这些协议包含了标准化的明显痕迹,它们包含了太多的可选择项,这样会使这些协议的简单思想变得含糊不清,因此我们不想在本书中介绍这些协议。

第二,这些协议已经有了认证协议的“正常规模”,这使得它们不适合作为构建高层认证协议的基本模块。另外,这些协议实际上包含了一些不太好的特点,比如要求协议的参与者维护序列号(包括 TTP,也就是有状态的 TTP!)。因此,这两个协议必然不适合作为未来协议设计者的协议构造模型!另外,在现实中使用这两个协议中的任何一个协议,都应该十分谨慎。

我们将介绍一个包含 TTP 的实体认证协议。然而,该协议是不安全,它容易受到将在以后介绍的多种攻击。

#### 11.4.3.1 Woo-Lam 协议

该协议由 Woo 和 Lam 设计[303]。因此,我们称之为 Woo-Lam 协议。协议 11.2 是对它的描述。

我们选择介绍 Woo-Lam 协议决不是要推荐该协议作为模板。相反,该协议不仅在许多方面有致命缺陷,并且虽然它还有几个不同的修正版本,然而这几个版本也都是有缺陷的。该协议还包含令人不满意的一些设计特性,我们将揭示、评论和识别在此协议中发现了诸多缺陷的



一个基本原因。所以我们认为,在设计正确认证协议的难题的研究中,Woo-Lam 协议扮演一个有用的角色。

该协议的目标在于,即使 Alice 和 Bob 开始互不相识,Alice 仍然能够向 Bob 证明自己。

### 协议 11.2 Woo-Lam 协议

前提:Alice 和 Trent 共享对称密钥  $K_{AT}$ ,

Bob 和 Trent 共享对称密钥  $K_{BT}$ ;

目标:Alice 向 Bob 证实她自己,虽然 Bob 还不认识 Alice。

1. Alice  $\rightarrow$  Bob: Alice;
2. Bob  $\rightarrow$  Alice:  $N_B$ ;
3. Alice  $\rightarrow$  Bob:  $\{N_B\}_{K_{AT}}$ ;
4. Bob  $\rightarrow$  Trent:  $\{Alice, \{N_B\}_{K_{AT}}\}_{K_{BT}}$ ;
5. Trent  $\rightarrow$  Bob:  $\{N_B\}_{K_{BT}}$ ;
6. Bob 使用密钥  $K_{BT}$  解密密文分组,如果解密后正确地返回 Bob 的一次性随机数,Bob 就接受此次运行,否则拒绝。

开始时,因为 Alice 和 Bob 互不相识,Alice 只能向 Trent 展示其加密能力:Alice 使用她和 Trent 共享的长期密钥加密 Bob 的一次性随机数  $N_B$  (步骤 3)。Trent 作为 TTP,将诚实地按照协议的要求解密 Alice 构造的密文(在步骤 4 接收到消息后)。最后,当 Bob 从密文分组中重新获得他的一次性随机数后,Bob 就可以断定:Trent 诚实的密码操作只能是在 Alice 的密码操作之后,并且这些操作都是在他认为新鲜的一次性随机数基础上进行;这样,就证明并确认了 Alice 的身份和活现性。

另外,Woo-Lam 协议可以看做是建立在 11.4.1.1 节中所介绍的标准协议结构的基础上。例如,第 2 行和第 3 行消息与机制(11.4.1)是类似的,第 3 行和第 4 行消息也是类似的。

我们将揭示 Woo-Lam 协议的几个安全缺陷推迟到 11.7 节。另外,该协议包含某个深层的令人不满意的设计缺陷,我们认为该缺陷是造成该协议安全缺陷的主要原因。然而,我们将进一步把我们对那一设计缺陷的分析和评论推迟到 17.2.1 节给出,在那里,我们将研究使用形式化的方法来获得正确的认证协议。

## 11.5 基于口令的认证

由于口令简单易记,在远程访问计算机系统的“用户-主机”模式中,广泛应用基于口令的认证。在这种类型的认证中,用户和主机共享某个口令,该口令实质上等价于一个长期然而相当短的对称密钥。

如果用户  $U$  希望使用主机  $H$  提供的服务, $H$  必须首先对  $U$  进行初始化并发给他一个口令。 $H$  保留一个存有所有用户口令的文档。该文档的每一行记录都是一对数据  $(ID_U, P_U)$ ,其中  $ID_U$  是  $U$  的身份, $P_U$  是  $U$  的口令。用于  $U$  访问  $H$  的一种直接的基于口令的协议可能为下面的形式:

1.  $U \rightarrow H: ID_U$ ;
2.  $H \rightarrow U: \text{“口令”}$ ;
3.  $U \rightarrow H: P_U$ ;
4.  $H$  从其口令文档中寻找记录  $(ID_U, P_U)$ , 如果接收的  $P_U$  与记录中的匹配, 就允许访问。

我们应该注意, 该协议实际上并未实现任何意义上的实体认证, 甚至连从  $U$  到  $H$  的单方认证也没有实现。这是因为该协议没有哪一部分涉及了用于鉴别  $U$  的真实通信的新鲜性标志符。然而, 在 20 世纪 70 年代初, 用户通过一个非智能终端接到主机, 在主机和终端之间的通信链路是不可攻击的专线, “口令认证”这一术语就是在这个时候开始使用。在这样一种设备和通信环境下, 上述协议的确提供了从  $U$  到  $H$  的单方认证。

然而, 在远程开放网络通信的环境下, 因为上面的口令协议中任何主体都没有执行密码操作, 所以会带来两个严重的问题。

第一个问题是主机  $H$  所维护的口令文件的脆弱性。因为保存的口令文件可能会被 Malice 读取(现在 Malice 是内部人员, 甚至是系统管理员)。在 Malice 拥有口令文件之后, 他就拥有了所有用户的所有权限; 于是他就能够通过伪装某个用户登录系统, 从而对该用户甚至整个系统造成不可检测的损害。很明显, 在某个用户名下攻击系统降低了 Malice 被发现的危险。

基于口令的简单远程访问协议带来的第二个问题是, 从  $U$  到  $H$  发送的口令是明文形式, 这样该口令可能会被 Malice 窃听。该攻击称为在线口令窃听。

### 11.5.1 Needham 口令认证协议及其在 UNIX 操作系统中的实现

Needham 首先提出了一个非常有效并且十分简单的方法, 该方法可以解决口令在主机中的安全存储问题(见[106]“致谢”部分或[134])。主机  $H$  可以使用某个单向函数来对口令编码, 也就是说, 记录  $(ID_U, P_U)$  由  $(ID_U, f(P_U))$  代替, 其中  $f$  是一个单向函数, 对该函数求逆是困难的。前面给出的简单“口令协议”也应该改为协议 11.3 的形式。

#### 协议 11.3 Needham 的口令认证协议

前提: 用户  $U$  和主机  $H$  已经设置了记录  $(ID_U, f(P_U))$ , 其中  $f$  是某个单向函数;

$U$  记住口令  $P_U$ ;

目标:  $U$  使用其口令登录到  $H$ 。

1.  $U \rightarrow H: ID_U$ ;
2.  $H \rightarrow U: \text{“输入口令”}$ ;
3.  $U \rightarrow H: P_U$ ;
4.  $H$  对  $P_U$  应用函数  $f$ , 然后从口令文档中找出记录  $(ID_U, f(P_U))$ , 如果计算得到的  $f(P_U)$  与记录中的数据匹配, 就允许访问。

协议 11.3 所实现的就是 UNIX<sup>①</sup>操作系统的口令认证协议。其中函数是用 DES 加密算法实现的(见 7.6 节)。系统在主机  $H$  中存储口令文件, 其中包含用户的身份 UID 和一条密文,

① UNIX 是贝尔实验室的商标。

该密文是对 64 个 0 组成的串(作为输入)进行加密变换生成的,变换使用了 DES 加密算法,算法的输入密钥为用户的口令  $P_U$ 。为了防止用市面上可以买到的高速 DES 硬件来破译口令,上述变换实际上并不是一个单纯的 DES 加密。事实上,该变换进行了 25 轮的 DES 加密,并加入了一种称为“比特-交换置换”的操作。“比特-交换置换”是对每一轮的输出密文进行的置换。在每一轮,DES 输出密文中的某些比特位根据某个 12 比特随机数进行交换,该随机数被称为盐(salt),也存储在口令文件中,该操作也可称为加盐操作。每一轮密文经过“比特-交换置换”后便作为 DES 下一轮加密的输入。该机制的详细情况见[208]。

这样,使用 DES 函数的变换  $f(P_U)$  可以看做是对常数串  $0^{64}$  进行的、带密钥的和参数化的单向杂凑函数。其中,密钥是口令  $P_U$ ,参数是盐。因为在这一过程中涉及了加盐操作,所以存储在  $H$  中的口令文件中的口令记录应该看做是  $(ID_U, salt, f(P_U, salt))$ 。然而为了叙述清晰,我们仍将使用  $f(P_U)$  代替  $f(P_U, salt)$ 。

这样,在 UNIX 上实现 Needham 口令认证协议之后,从主机  $H$  窃取口令文件  $f(P_U)$  来攻击系统,对于 Malice 来说就不再是行之有效的方法。首先,  $f(P_U)$  不能在协议 11.3 中直接使用,因为这样会使  $H$  计算  $f(f(P_U))$  从而使 Malice 不能通过检验。其次,对单向函数  $f$  求逆在计算上不可行,尤其当变换包含 25 轮和“比特-交换置换”之后,求逆更为困难。所以,如果用户选择了合适的口令,以致于该口令不容易被猜测,那么对于 Malice 来说从  $f(P_U)$  得到  $P_U$  就很困难了(我们将在 11.5.3 节中讨论猜测口令的问题)。

尽管口令文件的机密性不必再做过多考虑,口令文件的完整性仍然必须维护。同样,协议 11.3 容易受到在线口令窃听攻击。一次性口令机制就是为了抵抗该攻击而设计的,下面我们来描述该协议。

### 11.5.2 一次性口令机制(及缺陷的修补)

Lamport 提出了一个简单的思想来挫败在线的口令窃听[176]。该技术可以看做是一次性口令机制。这里,“一次性”的意思是  $U$  发送给  $H$  的口令不会重复,但是这些口令在计算上是相关的。这样,因为从协议的某次运行中窃听的口令以后不能再用,所以也就成功地防止了口令窃听。

在用户初始化时,  $U$  的口令记录设置为  $(ID_U, f^n(P_U))$ , 其中

$$f^n(P_U) \stackrel{\text{def}}{=} \underbrace{f(\cdots(f(P_U))\cdots)}_n$$

$n$  是一个大整数。同前面的口令认证协议中的一样,用户只须记住  $P_U$ 。

$U$  和  $H$  首次运行口令认证协议,在要求输入口令时(口令认证协议的第 2 行消息),  $U$  的计算设备,比如客户平台或计算器,会要求  $U$  输入  $P_U$ ,然后重复计算  $f$  函数  $n-1$  次,得到  $f^{n-1}(P_U)$ 。即使当  $n$  比较大(例如,  $n=1000$ )时也能够很有效地完成。计算结果会在认证协议的第 3 行发送给  $H$ 。

在接收到  $f^{n-1}(P_U)$  以后,  $H$  会对接收的口令执行一次  $f$  运算,得到  $f^n(P_U)$ ,然后执行口令认证协议第 4 步的正确性检验。如果通过检验,  $H$  就认为接收的值是  $f^{n-1}(P_U)$ ,并且是从  $P_U$  计算得到的,而该  $P_U$  是在初始化时设定的,因此通信对方必然是  $U$ 。这样,  $U$  就被允许进入该系统。另外,  $H$  将会更新  $U$  的口令记录:用  $f^{n-1}(P_U)$  替换  $f^n(P_U)$ 。

在下次运行该协议时,  $U$  和  $H$  会分别使用  $f^{n-2}(P_U)$  和  $f^{n-1}(P_U)$ , 就像前一次使用  $f^{n-1}(P_U)$  和  $f^n(P_U)$  一样。所以该协议是有状态的, 它使用了计数器从  $n$  递减到 1。当计数器的值是 1 时,  $U$  和  $H$  应该重新设置口令。

这种方法要求  $U$  和  $H$  在口令的状态上是同步的: 当  $H$  在  $f^i(P_U)$  状态时,  $U$  必须在处于  $f^{i-1}(P_U)$  状态。这种同步可能会丢失, 例如因为“不可靠”的通信链路或者主机系统“死机”。应该注意, 这里的“不可靠”和“死机”可能是 Malice 造成的。

Lamport 提出了一种简单的方法用于同步丢失时重建同步[176]。本质上, 该方法要求系统“向前跳”: 如果  $H$  在  $f^j(P_U)$  状态而用户在  $f^k(P_U)$  状态, 并且  $j \neq k+1$ , 那么同步就丢失了, 此时我们要求系统“向前跳”,  $H$  到达  $f^i(P_U)$  状态,  $U$  到达状态  $f^{i-1}(P_U)$  状态, 其中  $i \leq \min(j, k)$ 。很明显, 这种重新同步的机制需要  $H$  和  $U$  的双方认证通信。然而, 在 Lamport 很短的技术论文中没有给出这个必要性的细节。

Lamport 基于口令的远端访问机制已经被修补并实现为“一次性口令”系统, 称为 S/KEY<sup>①</sup>[136]。S/KEY 机制的修补是为了解决“不可靠通信”问题而提出的, 该修补要求  $H$  为  $U$  维护一个计数器。在用户初始化时,  $H$  保存用户  $U$  的口令记录  $(ID_U, f^n(P_U), c)$ , 其中  $c$  初始化为  $n$ 。协议 11.4 描述了 S/KEY 机制。

很明显, 在协议 11.4 中,  $U$  和  $H$  再也不会失去同步, 因此不可靠通信链路将不再是一个问题。

#### 协议 11.4 S/KEY 协议

前提: 用户  $U$  和主机  $H$  已经设定  $U$  的初始口令记录  $(ID_U, f^n(P_U), n)$ , 其中  $f$  是密码杂凑函数;

$U$  记住口令  $P_U$ ;

$H$  中  $U$  的当前口令记录是  $(ID_U, f^c(P_U), c)$ , 其中  $1 \leq c \leq n$ 。

目标:  $U$  向  $H$  认证同时不以明文形式传输  $P_U$ 。

1.  $U \rightarrow H: ID_U$ ;
2.  $H \rightarrow U: c$ , “输入口令”;
3.  $U \rightarrow H: Q = f^{c-1}(P_U)$ ;
4.  $H$  从其口令文档中查找记录  $(ID_U, f^c(P_U), c)$ ;

如果  $f(Q) = f^c(P_U)$ , 就允许接入, 并把  $U$  的口令记录更新为  $(ID_U, Q, c-1)$ 。

遗憾的是, S/KEY 对 Lamport 的原始协议的修改是件危险的事。我们注意到, 在最好情况下, 基于口令的远端访问协议所实现的也只是  $H$  识别  $U$  的身份。这样,  $H$  发送给  $U$  的计数器值可能事实上是由 Malice 发送的, 或者是已经被 Malice 修改过的。读者可以考虑 Malice 应该如何攻击, 例如怎样修改计数器的值, 以及如何跟着展开攻击。我们鼓励读者自己先攻击 S/KEY 协议, 而不要急于阅读 11.7.2 节。

<sup>①</sup> S/KEY 是 Bellcore 的商标。

也许有人会争论:“S/KEY 协议的安全性不比 Needham 口令认证协议(协议 11.3)的安全性低,因为后者的口令是以明文传输的!”然而,我们应该注意,Needham 的口令认证协议从来没有宣称它能够抵抗在线口令窃听攻击,而 S/KEY 协议的设计目的却包含了这一宣称。很不幸,S/KEY 没有实现这一目的。

### 11.5.3 加盐操作:加密的密钥交换(EKE)

大部分基于口令的系统建议用户选择 8 个键盘上的(ASCII)字符作为口令。因为这样的口令可以被大多数用户记住而不用写下来,同时因为每一个 ASCII 字符都由一个字节(8 比特)来表示,一个 8 字符口令就可以转化为一条 64 比特的字符串。64 比特串的空间有  $2^{64}$  个元素,这么大的空间是比较合适的。这样看起来,一个 8 字符口令应该能够防止对口令的猜测,甚至还能够防止非专业攻击者的穷举搜索攻击。

然而,“64 比特”口令并不是真的有 64 比特的信息量。尽管所有的 ASCII 字符的信息率并不是比 8 比特/字符小很多(见 3.8 节语言的信息率),然而人们通常并不用 ASCII 表中的随机字符作为口令。与此相反,通常都选择一些容易记忆但却不好的口令。典型的不好的口令就是字典中的单词或者人的名字,通常都是小写,可能以一两个阿拉伯数字结尾。香农估计英语的信息率在 1.0~1.5 比特/字符之间(见 3.8 节,这一估计基于所有小写的英文单词[267])。这样,8 字符口令的空间要远远小于  $2^{64}$ ,并且如果该空间中有相当一部分是不好的口令(小写字母单词、人名等),该空间就会更小。规模不大的口令空间就使得离线字典攻击成为可能。在这种攻击中,Malice 使用  $f(P_U)$  搜索一个包含不好口令的字典来匹配  $P_U$ 。因为该攻击是离线进行的,所以它可以是自动的,并能快速执行。这里我们应该注意 Lamport 的一次性口令机制也不能抗击离线口令攻击:Malice 可以窃听当前的状态值  $i$  和  $f^i(P_U)$ ,并进而展开字典搜索攻击。

Bellare 和 Merritt 提出了一个非常吸引人的协议,该协议可以实现安全的基于口令的认证。该协议称为加密的密钥交换(EKE)[30]。EKE 协议不仅可以抗击对于口令的在线窃听,还可以抗击离线字典攻击。本质上,EKE 机制中使用的技术是概率加密。在第 14 章,我们将研究概率加密的一般技术。这里,读者可以认为该技术是对口令进行了加盐操作。

与基于口令的认证协议(协议 11.3 和协议 11.4)不同,在那些协议中  $H$  只拥有  $U$  的口令的单向函数值,而在 EKE 协议中, $U$  和  $H$  共享口令  $P_U$ 。该共享口令用做对称密码密钥,正如前面我们所阐述的,该对称密钥可能是从一个相当小的口令空间选取的。

EKE 协议在协议 11.5 中给出。

EKE 协议的独创性在于前两个步骤。在步骤 1,密文组  $P_U(\mathcal{E}_U)$  是用口令  $P_U$  作为密钥,对一次性随机串  $\mathcal{E}_U$  加密得到的。在步骤 2,双重加密的密文组  $P_U(\mathcal{E}_U(K))$  中包含的是另一个一次性随机数:会话密钥  $K$ 。由于通常  $P_U$  是容易记忆的, $P_U$  是小空间其中的一个值,所以这两个随机串  $\mathcal{E}_U$  和  $K$  应该比  $P_U$  的规模大。这样,消息 1 和消息 2 的密文组就能够通过和  $P_U$  统计独立的方式把  $P_U$  隐藏起来。

我们必须强调,该协议中,正是这一次性  $\mathcal{E}_U$  的随机性才起到了“加盐操作”的作用。如果这个“公钥”不是一次性的,EKE 协议的独特功能就会完全丧失:该协议甚至可能帮助 Malice 搜寻口令  $P_U$ ,因为这时 Malice 能够利用教科书式公钥加密算法的弱点(例如,8.9 节的“中间相遇攻击”)。

**协议 11.5 加密的密钥交换(EKE)**

前提:用户  $U$  和主机  $H$  共享口令  $P_U$ ;

系统已经预先协商一种对称加密算法,  $K()$  表示以  $K$  为密钥的对称加密;  $U$  和  $H$  也已经协商一种非对称加密体制,  $\mathcal{E}_U$  表示以  $U$  为密钥的非对称加密。

目标:  $U$  和  $H$  完成双方实体认证, 并协商一个共享密钥。

1.  $U$  生成一个随机“公钥” $\mathcal{E}_U$ , 发送给  $H$ :

$$U, P_U(\mathcal{E}_U)$$

(\*“公钥”实际上并不是公开的, 它是非对称加密算法的加密密钥\*)

2.  $H$  用  $P_U$  解密密文组, 得到  $\mathcal{E}_U$ ;

$H$  生成随机对称密钥  $K$ , 并发送给  $U$ :

$$P_U(\mathcal{E}_U(K))$$

3.  $U$  解密双重加密的密文分组, 得到  $K$ ;

$U$  生成一次性随机数  $N_U$ , 并发送给  $H$ :

$$K(N_U)$$

4.  $H$  用  $K$  解密密文组, 生成一次性随机数  $N_H$ , 并发送给  $U$ :

$$K(N_U, N_H)$$

5.  $U$  使用  $K$  解密密文组, 并返回给  $H$ :

$$K(N_H)$$

6. 如果第 3、4、5 步的询问-应答是成功的, 就允许  $U$  登录, 通信双方用共享密钥  $K$  处理接下来的安全通信。

如果消息行 3、4、5 中加密的一次性随机数  $N_U$ 、 $N_H$  是随机生成的, 并且有足够大的规模 (例如, 比会话密钥  $K$  的规模大), 那么这些随机数就进一步隐蔽了会话密钥  $K$ , 其中的道理和前两条消息隐蔽  $P_U$  相同。这样,  $P_U$  依旧保持着和 EKE 协议中任何消息的统计独立性。

口令  $P_U$  和协议运行时传输的消息的这种统计独立性, 意味着该口令对于窃听者的安全是一种信息论意义上的安全 (见 7.5 节)。所以被动窃听者不能运用协议消息对  $P_U$  实施离线的口令攻击。攻击该协议的其他可能方法要么对  $P_U$  直接猜测, 要么通过修改协议消息实施主动攻击。考虑猜测攻击没有多大的意义, 因为不能阻止这种攻击方法。幸运的是, 这种方法通常是无效的。而主动攻击则会以很大的概率被诚实的协议参与者检测到, 并果断废止协议的运行。

在步骤 1 中  $U$  对随机公钥的加密以及步骤 2 中  $H$  对随机会话密钥的加密, 就是我们所称的对  $P_U$  的“加盐操作”。正是不断变化的“加盐操作”使得攻击者一无所获。因此, EKE 协议中前面的两行消息提供一种具独创性的技术。消息 3、4、5 实际上构成了传统上基于询问-应答机制的双方认证协议。事实上, 这几部分可以用基于对称密钥的双方认证协议结构代替。

EKE 协议很适合于采用 Diffie-Hellman 密钥交换体制来实现。设  $\alpha$  生成阶大于  $2^{64} > 2^{|P_U|}$  的一个群。在步骤 1,  $U$  的计算设备随机选取  $x \in (0, 2^{64})$ , 并计算  $\mathcal{E}_U = \alpha^x$ , 在步骤 2,  $H$  的计算



设备随机选取  $y \in (0, 2^{64})$ , 并计算  $\mathcal{E}_U(K) = \alpha^y$ 。  $U$  和  $H$  的共享会话密钥是  $K = \alpha^{xy}$ 。这样, 对于这个协商会话密钥, 每一方都有他自己的贡献。在实现时,  $U$  和  $H$  可以公开地协商群的生成元  $\alpha$ :  $U$  在预协商步骤中把关于群的描述(包括生成元  $\alpha$ )发送给  $H$ 。

注意, 我们仅仅要求  $\alpha$  生成的群的阶大于  $2^{64}$ 。这对于非对称密码系统群阶的下限来说是一个很小的数字。所以该协议的效率会很高。同时, 当群的阶很小时, 使得计算离散对数很容易, 从而解决计算 Diffie-Hellman 问题的难度降低。然而, 没有  $\alpha^x, \alpha^y, \alpha^{xy}$ , Diffie-Hellman 问题的难度降低对于寻找口令  $P_U$  是没有帮助的,  $P_U$  依旧在一个大小为  $2^{64}$  的空间中保持统计独立。同样, 如果消息 3、4、5 中加密的一次性随机数足够大并且是随机的, 会话密钥  $K$  就应该依旧在其阶大于  $2^{64}$  的群中保持统计独立。这样, 离线字典攻击和在线口令攻击仍然是困难的。

本质上, 对口令的随机“加盐操作”“放大了”口令的空间, 从口令的字典变成为某个随机的非对称密钥空间, 这就是 EKE 协议背后的技巧。

## 11.6 基于非对称密码学的认证密钥交换

如果协议输出的共享密钥来自参与协议的某一个主体, 我们就说该协议通过**密钥传输机制**建立了共享密钥。如果协议输出的共享密钥是某个函数的输出, 且该函数的输入是协议所有参与者的随机输入, 我们就说该协议通过**密钥交换(或密钥协商)**机制建立了共享密钥。与密钥传输机制相比, 密钥交换机制的优点在于共享秘密的各参与方对最后的输出都有其自己的控制权。因此, 对于输出密钥的质量, 各参与方都有比较高的信心。

除了采用 Diffie-Hellman 实现的 EKE 协议外, 迄今我们所介绍的基本认证技术中, 涉及到密钥建立的协议均使用密钥传输机制。接下来我们介绍密钥交换机制。

密钥交换可以通过伪随机函数或者单向杂凑函数的输出作为密钥来实现, 密钥共享的各方都拥有该函数的部分输入。最常使用的方法归功于 Diffie 和 Hellman 的伟大发现: Diffie-Hellman 密钥交换, 我们将其看做是一个单向函数(见 8.4 节的注释 8.1.3)。我们在协议 8.1 中已经描述了 Diffie-Hellman 交换。该机制在不使用加密的条件下, 在两个远程主体之间实现了密钥协商。

协议 8.1 是 Diffie-Hellman 密钥交换的基本形式, 它实现了无认证的密钥协商。在攻击 8.1 中, 我们介绍了**中间人攻击**, 攻击完成后, Malice 和 Alice 共享了一个密钥, 同时 Malice 也和 Bob 共享了另一个密钥, 因此, 他能够转发 Alice 和 Bob 之间的“机密”通信。Diffie-Hellman 密钥交换的正确应用必定是协议 8.1 的某种变形。最简单的变形是双方协议, 在这个协议中, Alice 预先知道  $g^b$  是 Bob 的公钥:

$$1. \text{ Alice} \rightarrow \text{Bob}: \text{Alice}, g^a \quad (11.6.1)$$

其中数值  $a$  是 Alice 从一个相当大的整数区间随机选取的。

在发送(11.6.1)中的消息后, Alice 就知道  $g^{ab}$  是仅由她和 Bob 共享的密钥。因为除了 Alice 和 Bob 之外, 任何人要求出  $g^{ab}$  相当于解决计算性 Diffie-Hellman 问题(CDH 问题, 见 8.4 节的定义 8.1), 而该问题被认为在计算上是困难的。另外, 因为 Alice 随机选择了某个新的  $a$  作为指数, 所以协商密钥对于 Alice 来说是新鲜的, 这就意味着该密钥对于 Alice 来说是认证的。然而, 接收到  $g^a$  以后, Bob 并不知道是谁和他共享密钥  $g^{ab}$ , 也不能确认该密钥是否是新鲜的。因而, 上面简单的变形协议只是实现了单方认证的密钥协商。

应用迄今所介绍的各种机制,我们不难将(11.6.1)的机制扩展,以获得双方认证的密钥协商。比如,Alice 可以对  $g^a$  和其身份以及时间戳进行数字签名。

下面我们介绍一个广为人知的认证的密钥交换协议,该协议也是 Diffie-Hellman 密钥交换的一种变形。

### 11.6.1 工作站-工作站协议

工作站-工作站(STS)协议由 Diffie 等提出[100]。

在 STS 协议中,Alice 和 Bob 应该预先就所用的由共同元素  $\alpha$  生成的高阶有限阿贝尔群达成一致。同一系统范围内的用户可以使用同一个生成元  $\alpha$ 。读者可以回顾 8.4.1 节关于在建立共享群时应注意的问题,在 STS 协议中将会用到这个共享群。

Alice 和 Bob 各自拥有公钥证书

$$\text{Cert}_A = \text{sig}_{CA}(\text{Alice}, P_A, \text{desc}(\alpha))$$

$$\text{Cert}_B = \text{sig}_{CA}(\text{Bob}, P_B, \text{desc}(\alpha))$$

其中 CA 是证书机构(见第 13 章)。  $P_A$  和  $P_B$  分别是 Alice 和 Bob 的公钥,  $\text{desc}(\alpha)$  是对  $\alpha$  生成的共享群的描述。另外,通信双方已经就所用的对称密钥加密算法达成一致,有关符号的使用我们将遵循定义 7.1(见 7.2 节)。加密算法也能够在同一系统范围的用户间达成一致。

STS 协议的描述如协议 11.6 所示。

#### 协议 11.6 工作站-工作站(STS)协议

前提: Alice 拥有她的公钥证书  $\text{Cert}_A$ , Bob 拥有他的公钥证书  $\text{Cert}_B$ , 同一系统范围的用户共享某个高阶有限阿贝尔群  $\text{desc}(\alpha)$ , 并且这些用户共同确认了某个对称加密算法  $\mathcal{E}$ ;

目标: Alice 和 Bob 实现双方认证和双方认证的密钥交换。

1. Alice 随机选取某个大整数  $x$ , 并发送以下消息给 Bob:

$$\alpha^x$$

2. Bob 随机选取某个大整数  $y$ , 并发送以下消息给 Alice:

$$\alpha^y, \text{Cert}_B, \mathcal{E}_K(\text{sig}_B(\alpha^y, \alpha^x))$$

3. Alice 给 Bob 发送以下消息:

$$\text{Cert}_A, \mathcal{E}_K(\text{sig}_A(\alpha^x, \alpha^y))$$

其中  $K = \alpha^{xy} = \alpha^{yx}$ 。

警告: 该协议存在微小缺陷; 在 11.6.3 节会给出分析。

力图使 STS 协议具有以下四个安全属性(只有将该协议中的某个小缺陷修正后, 其中的几个才能实现):

**双方实体认证** 如果按照 STS 协议作者给出的认证的严格定义, 那么实际上该协议就没有实现该属性。在[100]中, Diffie 等在这方面犯了两个错误, 我们将分别在 11.6.2 节和 11.6.3 节中讨论。

**双方认证的密钥协商** 密钥协商部分显然是以 Diffie-Hellman 密钥交换协议为基础的;协商密钥的新鲜性由双方各自适当地随机选取指数来保证;双方独享确认密钥这一点由双方对于密钥生成素材的数字签名来保证。然而,把所有这些特性放在一起实际上并不能达到双方认证的密钥协商:只有把该协议中的某个小缺陷修补后,这一性质才能满足。

**双方密钥确认** 协议结束后,双方都可以看到对方使用确认的密钥加密了生成该密钥的素材。另外,正确的双方密钥确认依赖于正确的双方认证,而正确的双方认证只有把 STS 协议的一个小缺陷修正后才能得到。

**完善前向保密** 这是密钥建立协议的一条很吸引人的性质。意思是即使在密钥建立协议中所使用的长期私钥在某个时间点泄漏了,在该时间点以前所建立的会话的安全性不会受到影响[135,100]。如果某个密钥建立协议中的会话密钥是通过使用 Diffie-Hellman 密钥交换机制正确协商获得的,那么该密钥建立协议就具有 PFS 特性。在 STS 协议中,长期密钥是 Alice 和 Bob 各自的私钥。因为在一次协议运行中所协商的会话密钥是两个短暂密钥经过单向函数作用后得到的结果,并且在协议运行结束之后,就把这两个短暂密钥安全地除去,因此,用来签名的长期密钥的泄漏并不会影响以前协商的会话密钥的秘密性。

**匿名性(可否认性)** 如果公钥证书在各自密文分组中被加密,那么该协议运行中所传输的消息就不会泄漏给任何涉及此次消息交换的第三方。然而我们应该注意到,通信协议底层传输的地址信息可能会泄漏协议参与者的身份。因此,精确地说,“匿名”应该改述为某种类型的“可否认性”。这里的“可否认性”是指网络的监控者不能证明一个给定的协议副本是在哪两个特定主体之间发生的。因为 STS 协议是用于因特网安全的因特网密钥交换协议(IKE)[137,160,227]的基础之一,所以该属性也是 IKE 的一个特性。我们将在下一章的 12.2 节中研究 IKE(和这一属性)。

STS 协议,尽管在协议 11.6 中给出的版本存在小小的缺陷,但是在认证和认证的密钥交换这一领域依然是重要的和有影响的工作。它是“因特网密钥交换(IKE)协议”[137,227]的基础之一,而 IKE 协议是用于因特网安全的工业标准认证协议。我们将在 12.2 节介绍 IKE,从中可以看到 STS 协议对它的影响。

文献[100]包含两个缺陷:一个比较严重的缺陷存在于 STS 协议的一个简化版本中,该简化版本是一个“惟认证”的用法。另一个微小的缺陷存在于 STS 协议中。如果该协议在设计时遵从了现在普遍认可的一个设计准则的话(该设计准则在[100]发表之后被论证并得到普遍认可),那么以上两个缺陷都不会存在。下面我们介绍两个缺陷。对两个缺陷的研究会引出普遍认可的设计准则。

### 11.6.2 简化 STS 协议的一个缺陷

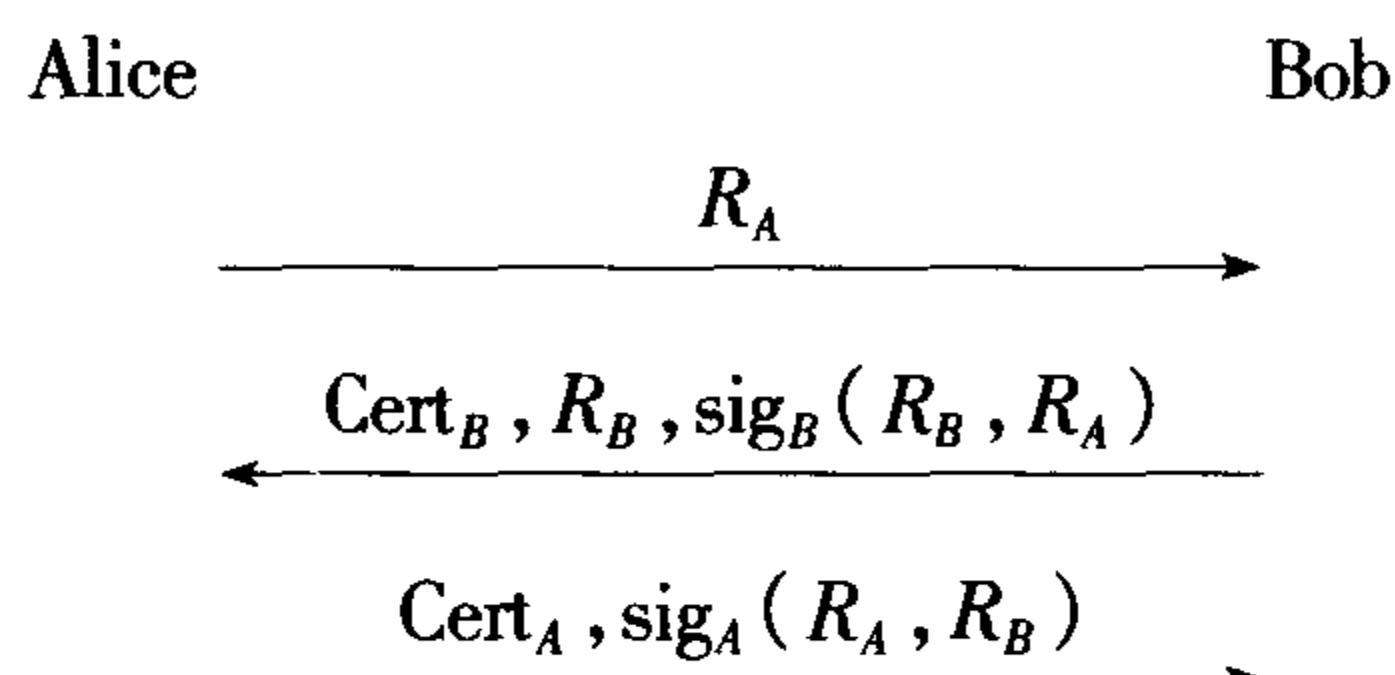
为了证明 STS 协议具有双方认证的属性,Diffie 等人对 STS 协议进行了简化,得到他们称之为“惟认证”的 STS 协议(见[100]的 5.3 节)。他们宣称“简化的协议和 ISO 提出的三次传输认证协议在本质上是相同的”。Diffie 等人所指的“ISO 协议”实际上就是我们所称的“ISO 公钥三次传输双方认证协议”(容易受到 Wiener 攻击的协议 11.1 的修正版,见 11.4.2 节)。

协议 11.7 所描述的就是“惟认证”STS 协议。

## 协议 11.7 有缺陷的“惟认证”STS 协议

前提: Alice 拥有她的公钥证书  $Cert_A$ 。

Bob 拥有他的公钥证书  $Cert_B$ 。



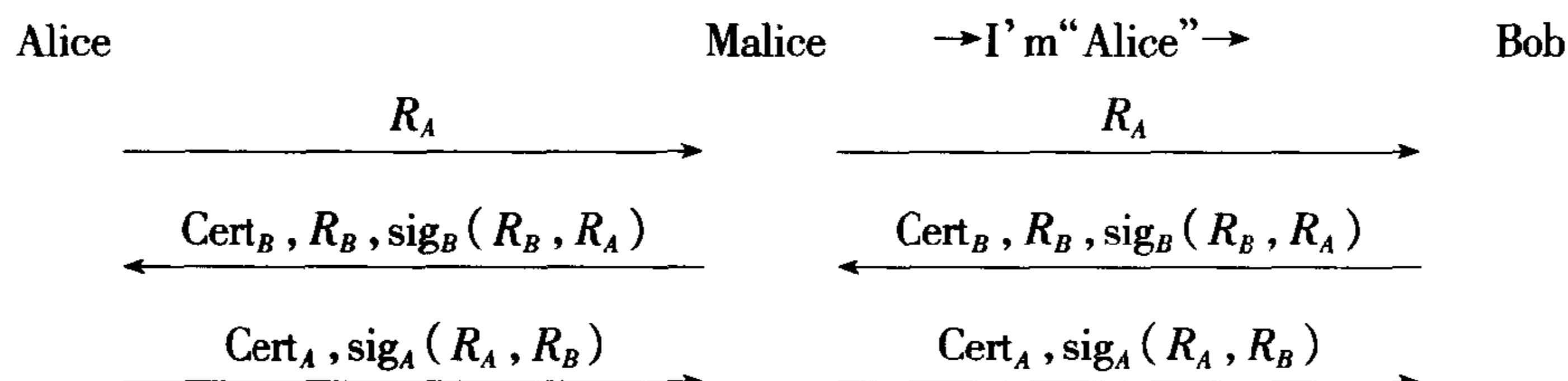
然而,协议 11.7 与 ISO 协议有一个重要的区别。在简化的 STS 协议中,签名消息没有包含协议参与者的身份,而在 ISO 协议中签名消息包含了参与者的身份。于是,简化的 STS 协议易受到“替换证书签名攻击”,我们在攻击 11.2 中已经给出了论证。

## 攻击 11.2 对于“惟认证”STS 协议的攻击

前提:除了协议 11.7 的前提之外,Malice 拥有证书  $Cert_M$ 。

(\* 所以 Malice 也是系统的合法用户 \*)

(\* Malice 面对 Alice 时使用其真实身份,但是在面对 Bob 时,他冒充 Alice \*)



后果: Bob 认为他和 Alice 进行了一次对话,而 Alice 认为它只是和 Malice 进行了一次对话。

该攻击中, Malice 也是系统的一个合法用户,因此也有一个公钥证书,然后他等着 Alice 发起对话。当这样一个机会出现以后,他就开始假冒 Alice 并使用 Alice 的随机数来发起和 Bob 的对话。在接收到 Bob 的响应以后, Malice 用他自己的证书和签名分别替换 Bob 响应中的证书和签名。这样做的结果是 Malice 成功地说服 Alice 对 Bob 的一次性随机数进行签名,并进而允许 Malice 成功地欺骗 Bob。该攻击是一次完美的攻击,因为 Alice 或者 Bob 都不能觉察到任何错误。

注意到在 Malice 所构造的整个攻击流程中,他并不是简单地被动窃听:他对 Bob 的随机数进行签名,并因此成功地说服 Alice 对 Bob 的随机数签名,进而成功地欺骗了 Bob。如果, Malice 只是被动的,例如,表现得像一根电线,那么 Bob 不可能接收到 Alice 对 Bob 的签名,并因此不会受骗。

上面的“替换证书签字攻击”对于 STS 协议并不适用,因为未简化版本中采取的加密操作使得 Malice 不能替换 Bob 的签名。该攻击对于 ISO 协议(协议 11.1)也不适用,因为 Alice 的签名将包含 Malice 的身份,所以这样的签名消息不能欺骗 Bob。

这里比较有趣的一点是,Diffie 等在他们的论文中(见[100]的 5.1 节)讨论了对已舍弃的简化 STS 协议的攻击,这里的简化 STS 协议去掉了加密过程,而这个攻击类似于“替换证书签名攻击”。然而,该攻击并没有对“惟认证”STS 协议进行攻击试验,这可能是因为“惟认证”STS 协议在形式上与修正后的 ISO 协议相似的缘故。同一篇论文中(见[100]的第 6 节)也讨论了对于有缺陷版本的 ISO 协议的 Wiener 攻击。Wiener 攻击显然与“替换证书签名攻击”不同,读者可以验证 Wiener 对于有缺陷的 ISO 协议的攻击应用于“惟认证”STS 协议时是无效的。由前面的分析,我们看到了认证协议的易出错本质。

在签名中包含验证者的身份确实是解决这类缺陷的一个有效方法。然而,这并不说明解决该缺陷的惟一方法就是包含验证者的身份,在一些应用中(例如,“因特网密钥交换(IKE)协议”,见 12.2.3 节),为了获得隐私的属性,不能包含协议参与者的身份信息(见 12.2.4 节)。这时,使用一种新的密码原型可以得到一种新的方法,既能保留隐私的属性,同时也能够解决上述缺陷。这一点我们将在以后的章节介绍。

### 11.6.3 STS 协议的一个瑕疵

Lowe 在[181]中发现了对 STS 协议的一种危害不大的攻击。在给出 Lowe 的攻击之前,我们先介绍关于认证的一种严格定义,该定义由 STS 的作者给出。

在[100]中,Diffie 等用“运行的匹配记录”概念定义了认证协议的安全运行。假设每一个协议参与者都记录在协议运行中所接收的消息。那么“一次运行的匹配记录”是指由某一主体所生成的消息必须在另一参与主体的记录中按照该消息发送时的相同顺序出现,反之亦然。认证协议的不安全运行定义如下([100]的定义 1):

如果参与协议运行的任何一方,例如 Alice,诚实地执行了协议,并接受了另一方的身份,而此时下面的条件成立: Alice 接受了对方的身份后,对方关于该运行的部分或全部记录与 Alice 的记录不匹配。

在这个认证协议不安全运行的定义下,在攻击 11.3 中展示的攻击就确实成为了一种攻击,虽然该攻击造成的损害很有限。

Lowe 的攻击在以下两个意义上说是一种小小的攻击:

- i) 在 Alice 和 Malice 所运行的这部分协议中,虽然 Malice 成功地欺骗了 Alice,但是 Malice 并不知道共享的会话密钥,因而在协议运行完之后 Malice 不能进一步欺骗 Alice。
- ii) 在 Bob 和 Malice 所运行的这部分协议中, Malice 没有完成协议,所以这一部分不是一个成功的攻击。

然而,我们说 Lowe 的攻击确实可以称为一种攻击,也有两个理由:

- I) Alice 接受 Bob 的身份信息是因为 Malice 只是简单地把 Bob 的消息逐比特地送给 Alice。而在 Bob 一方,因为 Bob 在对 Alice 的随机询问签名时,把通信对方看做是 Malice,所以 Bob 的通信记录就与 Alice 的不同。因此,这一攻击符合 STS 作者定义的“不安全运行”标准,也就是说,双方认证是失败的。诚然,实体认证这一概念本身就是很难精确刻划的,并且该领域是通过认证中的错误来学习和研究的,对某个攻击的认定如果是建立在早期的定义(例如,Diffie 等在[100]中给出的“不安全运行”)基础上,那么该认定就不足

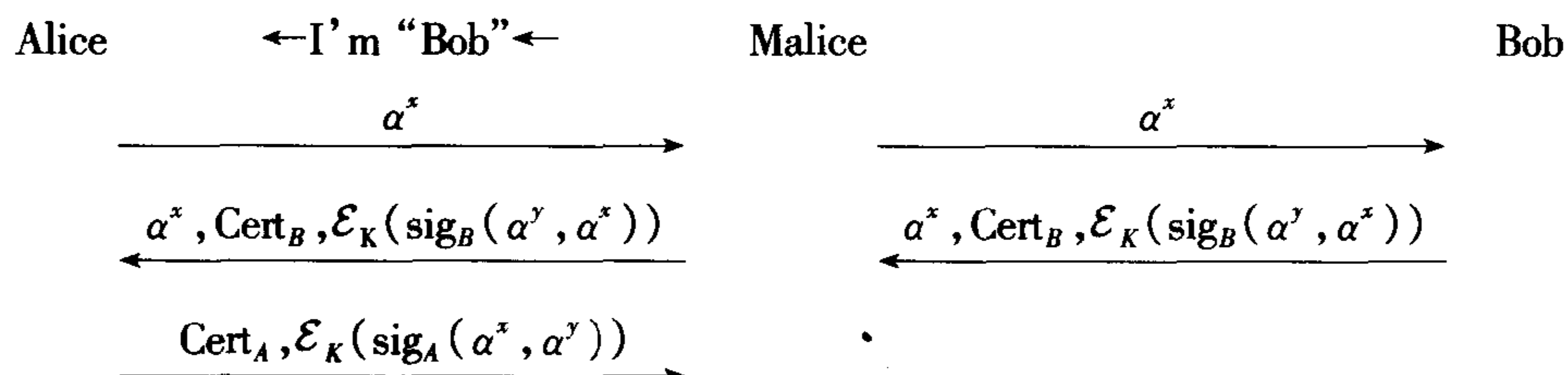


以令人信服。今天,人们会质疑早期的定义是否确实正确。然而,一种更好点的提问是“这些‘攻击’在今天从实际来看值得考虑吗?”在(II)中回答这个问题。

II) Malice 成功地欺骗 Alice,使她认为她在和 Bob 正常通信。然而,因为 Bob 从来就没有认为他是在和 Alice 通信,因此 Malice 接下来对 Bob 发送的请求或者为与 Bob 进行安全通信而做的准备都会被 Bob 拒绝,并且得不到任何解释。并且,也没有人通知 Alice 这种异常。我们可以把这种结果和另外一种意义不大的“攻击”进行比较,这种意义不大的“攻击”是指 Malice 一直是被动的,只是最后把一条 Alice 发送给 Bob 的消息截断了,使得 Bob 不能接收到最后一条消息。在这种意义不大的攻击中,为了获得匹配的记录,Bob 可能会通知 Alice 丢失了最后一条消息。今天在实际中,关于 Lowe 的攻击是否应该认为是攻击的问题,我们可以把 Alice 看做是中央服务器,可能会受到分布式攻击(例如,受到分布在网络上的 Malice 团伙的大规模攻击),此时,缺少终端用户(例如,许多的 Bob)的通知消息是确实要考虑的一个问题:因为服务器此时可能会为每一个终端用户保留资源,这样,服务器的服务能力就会极大的下降。我们应该特别指出的是,在 Lowe 的攻击中,Malice 和他的合作者们没有使用密码学的信任证件(证书),所以该攻击对于 Malice 来说开销很小。这和传统的拒绝服务攻击有很大的不同,在传统的拒绝服务攻击中,Malice 连同他的合作者和 Alice 对话时必须使用真实名称(例如,使用证书)

### 攻击 11.3 Lowe 对于 STS 协议的攻击(小小缺陷)

( \* Malice 面对 Bob 时使用他的真实身份,而在面对 Alice 时便假冒 Bob \* )



后果: Alice 被完全欺骗,她认为她和 Bob 进行了一次会话,并共享了某个会话密钥,而 Bob 认为他是和 Malice 运行了一次不完全的协议。Alice 决不会得到任何异常的通知,并且 Alice 接下来向 Bob 发送的请求或者为与 Bob 进行安全通信准备的资源都会被 Bob 拒绝,并且得不到任何解释。

基于(II)中解释的原因,我们称 Lowe 的攻击是对 Alice 实施的完善拒绝服务攻击,因为攻击者成功地使用了别人的密码证件。

如果修改该协议,使之符合由 Abadi 和 Needham 提出的一个正确且被广泛认同的协议设计准则[1],那么就可以成功地避免上面的攻击。该准则如下:

如果主体的身份信息对于该消息的含义是至关重要的,那么在消息中明确包含该主体的名字应当谨慎对待。

事实上,在 STS 协议中,签名消息应该包含协议参与双方的身份! 这样,Bob 发送和签署的消息中将含有 Malice 的名字“Malice”,从而该消息就不能再转发给 Alice 用于 Malice 欺骗 Alice。



另外,如果简化的“惟认证”STS 协议是由“签名中含有身份”的 STS 协议版本简化得到的,那么它也不会受到“替换证书签名攻击”,这是因为简化版本质上是 ISO 协议(协议 11.1)。

我们在前面已经提到,STS 协议是“因特网密钥交换(IKE)协议”[137,160,227]的基础之一。因此,在 12.2 节中我们会看到“完善的拒绝服务攻击”也能用于 IKE 的几个运行模式中。

最后,我们再次指出 11.6.2 节中阐述的一点:增加签名验证者的身份并不是防止这一攻击的惟一方法。例如,使用指定验证者签名可以在不增加身份信息的情况下更好地修正协议。我们将在后面的章节讨论这种方法。

## 11.7 对认证协议的典型攻击

在 2.3 节我们已经认识到 Malice(可能通过和他在分布网络上的合作者合作)可以在开放网络中窃听、截获、修改和注入消息,并且他通过冒充其他主体很善于做这类事情。从通信协议栈高层(应用层)来看,Malice 发动这些攻击的能力就像耍魔术一样:Malice 怎会如此的本事?

然而,从协议的底层通信(网络层)来看,Malice 发起这些攻击并不需要很复杂的技术。在 12.2 节我们将会看到这些技术,进而知道如何在底层通信协议上实现这些攻击,在那里我们还会了解网络层通信是如何进行的。目前,我们可以只是认为 Malice 具有魔术般的能力。Malice 可以对存在缺陷的协议发起各种攻击。

虽然我们不可能全部了解 Malice 使用的协议攻击技术(由于 Malice 可以发明新的攻击技术),但是知道一些典型的攻击技术可以帮助我们了解如何设计强安全的协议以防止这些攻击。这一节,让我们来看一看 Malice 的代表作中几个著名的协议攻击技术。应该注意到,虽然我们分类介绍这些攻击技术,然而在实际当中,攻击者往往同时会使用多种技术,东一点西一点,直到找到可攻击的方法。

在介绍之前,我们首先强调以下内容:

**注释 11.3** 对认证协议或者认证的密钥建立协议的成功攻击,通常并不是指攻破该协议所用的密码算法,如运用基于复杂性理论的密码分析技术。相反,它通常是指 Malice 能够以某种未授权并且不被察觉的方式获得某种密码信任证件或者破坏某种密码服务,同时不用攻破密码算法。这当然是因为协议设计的错误,而不是密码算法的问题。□

### 11.7.1 消息重放攻击

在消息重放攻击中,Malice 预先记录某个协议先前的某次运行中的某条消息,然后在协议新的运行中重放记录的消息。由于认证协议的目标是建立通信方之间的真实通信,并且该目标通常通过在两个或多个通信方之间交换新鲜的消息来实现,所以认证协议中的消息重放违反了认证的目标。

在 2.6.4.2 节,我们介绍了对 Needham-Schroeder 对称密钥认证协议重放攻击的实例(回顾攻击 2.2)。注意在那里(回顾 2.6.4.2 节的最后一段)我们只讨论了重放攻击的危险之一:重放的消息包含某个旧的被攻破的会话密钥(如 2.5 节中讨论的,Malice 已经发现了该值,可能是因为某个主体粗心地处理或者由于该会话密钥的其他脆弱性)。

该攻击还会产生另一个后果,该后果可能更为严重,可以称为认证失败,也就是在通信双方之间不存在真实通信。实际上,Malice 展开攻击(回顾攻击 2.2)并不需要等到 Alice 发起和

Bob 的通信以后再进行; Malice 可以跳过前两行, 在第 3 行重发记录的消息发起攻击, 当然前提是 Malice 知道  $K'$ :

3. Malice("Alice") $\rightarrow$ Bob:  $\{K', Alice\}_{K_{BT}}$ ;
4. Bob $\rightarrow$ Malice("Alice"):  $\{I'm Bob! N_B\}_{K'}$ ;
5. Malice("Alice") $\rightarrow$ Bob:  $\{I'm Alice! N_B - 1\}_{K'}$ 。

现在 Bob 认为 Alice 在和他通信, 而实际上 Alice 可能根本不在线。

消息重放攻击是对认证协议和认证的密钥建立协议的传统攻击。似乎我们对该攻击已经有了足够的警觉性。在 11.4 节介绍的基本和标准协议结构中普遍包含了新鲜性标志符(一次性随机数、时戳)。然而, 好的警觉性未必就意味着我们能够很好地防范这种攻击。认证协议的一个微妙之处在于, 即使协议设计者清楚地认识到不同背景下协议中的错误, 协议设计者还是会重复地犯错误。我们看下面的例子, 它是另一种形式的消息重放攻击。

在[295]中, Varadharajan 等提出了几个“代理协议”。利用代理协议, 一个主体可将他对另一个主体的信任传递给另外一些相信他的主体。在协议中, 客户端 Bob 和认证服务器 Trent 共享密钥  $K_{BT}$ 。Bob 生成时戳  $T_B$ , 并希望建立密钥  $K_{BS}$  与另一服务器  $S$  进行安全通信。而后,  $S$  构造  $\{T_B + 1\}_{K_{BS}}$ , 并发送以下消息:

5.  $S \rightarrow$  Bob:  $S, B, \{T_B + 1\}_{K_{BS}}, \{K_{BS}\}_{K_{BT}}$ 。

原文中作者推理:

Bob 得到  $K_{BS}$  以后, 可以使用  $T_B$  来验证  $S$  回送的消息(包含  $S$  对  $T_B$  的应答)的新鲜性, 所以会话密钥确实是新鲜的。

然而, 尽管新鲜性标志符是用  $K_{BS}$  加密过的, 但 Bob 不能获得任何关于  $K_{BS}$  新鲜性的保证。Bob 只能推断出  $K_{BS}$  最近被使用过, 但是  $K_{BS}$  可能是一个旧的密钥, 甚至是已经泄漏的密钥。

所以我们有如下注释。

**注释 11.4** 有时, 新鲜性标志符和消息间的密码综合只能说明该密码操作是新鲜的, 而不能说明被综合的消息是新鲜的。□

### 11.7.2 中间人攻击

中间人攻击本质上就是广为人知的“象棋大师问题”<sup>①</sup>, 它适应于缺少双方认证的通信协议。在攻击时, Malice 能够把协议的某参与者所提出的困难问题提交给另外的参与者来回答, 然后把答案(可能经过简单处理)交给提问的主体, 反之亦然。

在 2.6.6.3 节和 8.3.1 节, 我们已经介绍了中间人攻击的实例。其中一个是对 Needham-Schroeder 公钥认证协议的攻击, 另一个是对无认证的 Diffie-Hellman 密钥交换协议的攻击。

① 某个新手和远距离通信的两位象棋大师同时进行两场象棋赛, 在一场棋赛中执黑, 另一场棋赛中执白。每次她都是把她在某一场棋赛中对手的走子原封不动地用在另一场棋赛中, 这样她就能够保证自己要么同时和两位象棋大师下和, 要么输一场赢一场, 这位新手就通过这种作弊的方法来提升自己的象棋等级。

对 S/KEY 协议的中间人攻击(协议 11.4, 攻击 11.4)是关于 Malice 在没有攻破方案中使用的密码算法的情况下, 却能获得密码证件的另一个很好例证。

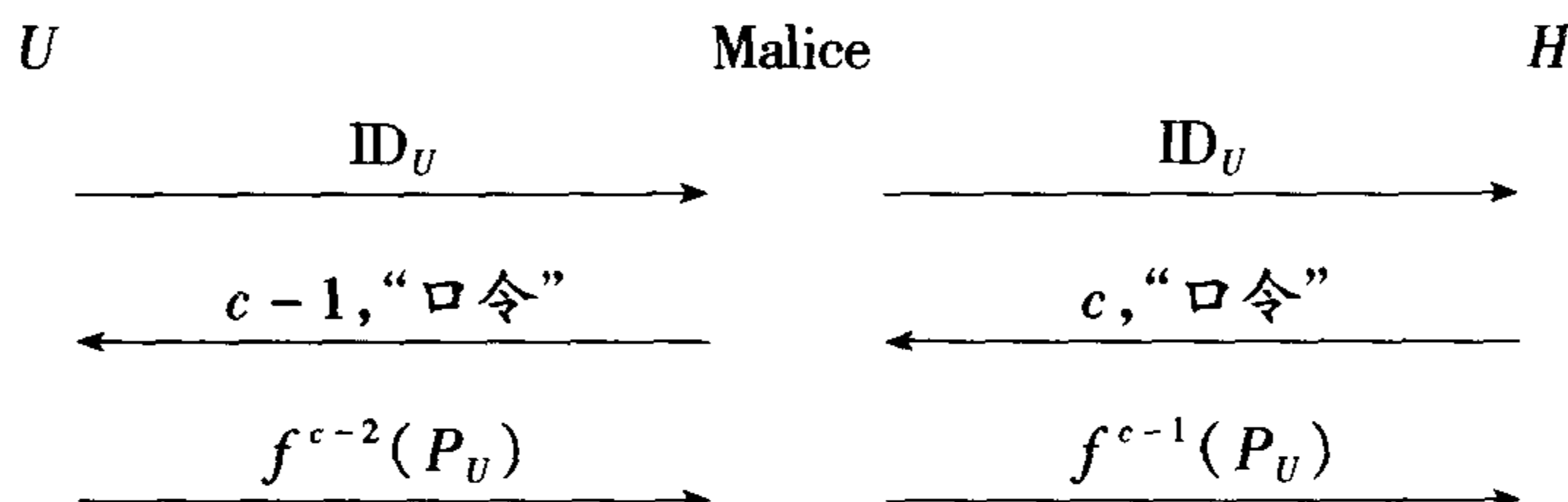
在 S/KEY 协议中使用的密码杂凑函数  $f$  可以是强安全的, 使得求逆在计算上是不可行的; 并且用户  $U$  也选择了合适的口令  $P_U$ , 使得执行离线字典攻击, 从  $f^c(P_U)$  找出  $P_U$  是不可行的(回顾 11.5.3 节离线字典攻击)。然而, 该协议在攻击 11.4 所示的主动攻击下依然会惨败。

攻击 11.4 成功的原因在于 S/KEY 协议中  $H$  给  $U$  的消息对  $U$  来说是没有认证性的。

对付中间人攻击的方法是在消息交换的两个方向上都提供数据源认证服务。

#### 攻击 11.4 对于 S/KEY 协议的攻击

(\* 该攻击中所用符号与协议 11.4 中的符号相同 \*)



后果: Malice 获得了  $f^{c-2}(P_U)$ , 在下一次会话时, 他就能以用户  $U$  的名义登录了。

### 11.7.3 平行会话攻击

在平行会话攻击中, 在 Malice 的特意安排下, 一个协议的两个或更多的运行并发执行。并发的多个运行使得 Malice 能够从一个运行中得到另外某个运行中困难问题的答案。

Abadi 和 Needham[1] 早期发现的对于 Woo-Lam 协议(协议 11.2)的攻击所使用的就是平行会话攻击。此攻击如攻击 11.5 所示。

如果 Bob 愿意几乎同时和 Alice 及 Malice 对话, 那么该攻击就是可行的。Malice 阻止到 Alice 的所有消息。在消息 1 和 1', Bob 被要求对两个运行做出响应, 一个运行是与 Malice 的, 另一个运行是与“Alice”的。在消息 2 和 2', Bob 分别响应两个不同的一次性询问随机数, 当然 Malice 会收到这两个随机数(其中的一个  $N_B$  是通过截获得到的), 然后 Malice 抛弃让他用的  $N'_B$ , 而使用给 Alice 用的  $N_B$ 。这样, 在消息 3 和 3' 中, Bob 接收到  $\{N_B\}_{K_{MT}}$ 。注意消息 3 和 3' 的密文可能是相同或不同的, 这取决于具体使用的加密算法(见第 14 章和第 15 章)。不管怎么样, Bob 所能做的只是分别在消息 4 和 4' 中简单地遵循协议规定: 对消息 3/3' 连同同一个目标主体的名字 Alice/Malice 进行简单的加密, 然后送给 Trent。注意, 即使 Bob 在消息 3 和 3' 中接收的密文分组是相同的(例如加密算法用是确定的, 尽管这样的加密算法今天的应用中已不多见), Bob 也不能注意到这一点, 因为密文分组对于 Bob 来说是不可识别的, 它不在 Bob 处理的消息范围之内。不处理“外界的密文”与我们对于诚实主体的约定(见 11.3 节)是相符的: Bob 没有参与攻击, 因此不能识别协议没有让他解密的密文分组。我们知道这样的表现很愚笨, 但是按照约定 Bob 应该就是这样“愚笨”的。在消息 5 和 5' 中, Trent 会正确地把  $N_B$  返回给 Bob, 从而使 Bob 相信“和 Alice 之间有一个好的运行”, 然而 Alice 根本不在线; 另一个密文解密后的

结果被认为是“垃圾”，因为它是 Trent 使用  $K_{AT}$  对  $\{N_B\}_{K_{MT}}$  解密的结果。最后结果是，Bob 拒绝和 Malice 的运行，接受了和“Alice”的运行。

在平行会话攻击中，两个平行的会话顺序并不重要。例如，如果 Bob 首先接收到 3'，然后接收到 3，攻击和原来一样。在 12.2 节中将看到，Bob 可以从网络层的地址信息来区分消息的来源。

Abadi 和 Needham 建议了对 Woo-Lam 协议的一种修改方法，这种方法我们稍后介绍。他们 also 把攻击 11.5 中的攻击通知了 Woo 和 Lam[1]。Woo 和 Lam 也提出了一系列的修改方法 [304]，包括 Abadi 和 Needham 的建议 ([304] 中称为  $II^3$ ) 和其他一些修改方法。最大的修改是  $II^4$ ：该修改要求把两个主体的身份信息，如 Alice 和 Bob，增加到所有的密文分组中。他们宣称其修改是安全的。遗憾的是，没有一个修改结果是安全的（包括 Abadi 和 Needham 的建议）。他们的每一个修改结果都可以用我们下面介绍的攻击类型来攻击。

### 攻击 11.5 对 Woo-Lam 协议的平行会话攻击

前提：除了协议 11.2 的前提外，Malice 和 Trent 共享长期密钥  $K_{MT}$ 。

( \* 所以恶意者 Malice 也是系统的某个合法用户 \* )

1. Malice(“Alice”)  $\rightarrow$  Bob: Alice;
  - 1'. Malice  $\rightarrow$  Bob: Malice;
  2. Bob  $\rightarrow$  Malice(“Alice”):  $N_B$ ;
  - 2'. Bob  $\rightarrow$  Malice:  $N'_B$ ;
  3. Malice(“Alice”)  $\rightarrow$  Bob:  $\{N_B\}_{K_{MT}}$ ;
  - 3'. Malice  $\rightarrow$  Bob:  $\{N_B\}_{K_{MT}}$ ;
  4. Bob  $\rightarrow$  Trent:  $\{Alice, \{N_B\}_{K_{MT}}\}_{K_{BT}}$ ;
  - 4'. Bob  $\rightarrow$  Trent:  $\{Malice, \{N_B\}_{K_{MT}}\}_{K_{BT}}$ ;
  5. Trent  $\rightarrow$  Bob: {“垃圾”}  $_{K_{BT}}$ ;
- ( \* 产生“垃圾”是因为 Trent 使用  $K_{AT}$  对密文组  $\{N_B\}_{K_{MT}}$  进行了解密 \* )
- 5'. Trent  $\rightarrow$  Bob:  $\{N_B\}_{K_{BT}}$ ;
  6. Bob 拒绝和 Malice 的运行;
- ( \* 因为解密后返回的是“垃圾”而不是一次性随机数  $N'_B$  \* )
- 6'. Bob 接受“和 Alice 的运行”，但实际上是和 Malice 的运行;
- ( \* 因为解密返回的  $N_B$  是正确的 \* )

后果：Bob 相信在协议的运行中是 Alice 在和他通信，而实际上 Alice 根本没有参与此次协议运行。

### 11.7.4 反射攻击

在反射攻击中，当一个诚实主体给某个意定的通信方发送消息用来让他完成密码操作时，Malice 截获该消息，并把该消息发送给消息的产生者。注意，这里把消息返回，并不是把消息

原封不动地返回, Malice 将会修改底层通信协议处理的地址和身份信息, 以便消息的产生者不会意识到反射回来的消息是由他“自己产生”的。我们将在 12.2 节介绍攻击者如何反射消息。

在此类攻击中, Malice 的目的是使消息的产生者相信反射过来的消息来自于消息产生者所意定的通信方, 该消息可以是对消息产生者的应答或者询问。如果 Malice 成功了, 那么要么消息产生者接受对问题的“回答”, 实质上是自问自答, 要么给攻击者提供了预言机服务, 完成了 Malice 所完不成的功能并把结果提交给 Malice。

对于 Woo-Lam 协议(协议 11.2), 在发现存在平行会话攻击(攻击 11.5)之后, Abadi 和 Needham 提出修改建议[1]: 在最后一条 Trent 给 Bob 的消息中增加 Alice 的身份:

$$5. \text{Trent} \rightarrow \text{Bob}: \{ \text{Alice}, N_B \}_{K_{BT}} \quad (11.7.1)$$

这样修改之后, 确实能够消除 11.5 所示的平行会话攻击, 因为攻击者再次进行平行会话攻击, 消息 5 会如下所示:

$$5. \text{Trent} \rightarrow \text{Bob}: \{ \text{Malice}, N_B \}_{K_{BT}}$$

而此时 Bob 期望的是(11.7.1), 这样就会检测到这种攻击。

然而, 尽管在协议中明确规定协议参与者的身份, 这无疑是设计安全认证协议的一条重要和谨慎的原则(在 11.7.7 节的另一种攻击类型中要讨论的问题), 但这也只是设计协议时诸多要考虑因素中的一个而已。常常是, 一种对策防范了一种攻击, 却会引入另一种攻击。Abadi 和 Needham 在文[1]中提出的 Woo-Lam 协议仍然是不安全的。修正的版本会受到攻击 11.6 所示的反射攻击(Clark 和 Jacob 在文[78]中提出)。

#### 攻击 11.6 对于 Woo-Lam 协议“修正”版本的反射攻击

前提: 与协议 11.2 的设定相同。

1. Malice(“Alice”)  $\rightarrow$  Bob: Alice;
2. Bob  $\rightarrow$  Malice(“Alice”):  $N_B$ ;
3. Malice(“Alice”)  $\rightarrow$  Bob:  $N_B$ ;
4. Bob  $\rightarrow$  Malice(“Trent”):  $\{ \text{Alice}, \text{Bob}, N_B \}_{K_{BT}}$ ;
5. Malice(“Trent”)  $\rightarrow$  Bob:  $\{ \text{Alice}, \text{Bob}, N_B \}_{K_{BT}}$ ;
6. Bob 接受。

后果: Bob 认为 Alice 在当前协议运行时是真实的, 而事实上 Alice 根本没有参与那次运行。

这里, Malice 执行了两次反射攻击: 消息 3 是消息 2 的反射, 消息 5 是消息 4 的反射。该攻击在这样一种假设下成立: 对于消息 3 和消息 5, Bob 在接收以后, 不会检测到任何不妥。而事实上, 根据我们对诚实主体的约定(见 11.3 节), 该假设是成立的。首先, Bob 在消息 3 中接收的随机分组确实是 Bob 产生并在消息 2 中发出的一次性随机数, 然而, Bob 却只能把消息 3 看做是不可识别的密文, 并且根据协议的规定, Bob 对消息 3 不能执行任何操作。同样, Bob 在消息 5 中接收的密文的确是他在消息 4 中产生并发出的, 然而, 对于消息对 4 和 5 来说, Bob 是无状态的。这和我们在 11.3 节中约定的无状态是相符的。因此, Bob 不能察觉到这样的攻击。



由 Woo-Lam 在文[304]中提出的一系列的修改也有类似的缺陷:都易遭受不同方式的反射攻击。对于最强的修改版  $\Pi^f$ , 每一条密文都包含双方的身份信息, 但如果 Bob 对不可识别密文组的大小不敏感, 反射攻击仍然能够成功。而根据我们对诚实主体的“愚笨”约定, Bob 不能察觉密文大小的变化的假设是合理的。

关于 Woo-Lam 协议及其各种修改版都存在缺陷的一个更为基本的原因将在 17.2.1 节研究, 在那里我们用形式化方法来设计正确的认证协议。在 17.2.2 节我们将推荐用来描述认证协议的一种正确方法。该方法将引出对于 Woo-Lam 协议和其他许多协议的一般修正版本。在 17.2.3.2 节, 我们将看到 Woo-Lam 协议的一般修改版(协议 17.2)能够抵抗目前所给出的任何攻击。

### 11.7.5 交错攻击

在交错攻击中, 某个协议的两次或多次运行在 Malice 的特意安排下按交织的方式执行。在这样一种攻击下, Malice 可以合成某条消息并发给某个运行中的某个主体, 期望收到该主体的一个应答; 而该应答可能对于另外某个运行中的另一个主体是有用的; 在接下来的运行中, 从前面运行中得到的应答可能会促使后面的主体对某个问题作出应答, 而这个应答又恰好能用于第一个运行, 如此交错地运行。

有些作者, 例如[35]的作者, 认为交错攻击是对前面两种攻击类型即平行会话攻击和反射攻击凑在一起的取名。然而, 我们认为这几种攻击类型是不同的, 交错攻击比平行会话攻击和反射攻击更为复杂。为了完成一次交错攻击, Malice 必须利用不同运行中消息的顺序关系。

Wiener 攻击(攻击 11.1)是交错攻击的一个很好的例证, 它是 Wiener 对“ISO 公钥三次传输双方认证”协议早期草案的攻击, 我们在 11.4.2 节已经介绍。在该攻击中, Malice 冒充  $B$  发起同  $A$  的一次协议运行(消息行 1); 收到  $A$  的应答(消息行 2)后, Malice 冒充  $A$  发起同  $B$  的一次新的协议运行(消息 1');  $B$  的应答(消息行 2')为 Malice 提供了  $A$  正在等待的应答。这样, Malice 就完成了和  $A$  的一次协议运行。与平行会话攻击(例如, 攻击 11.5)相比, 交错攻击对于消息交换的顺序是敏感的。

与 Wiener 攻击类似, 对于“惟认证”STS 协议的“替换证书签名”攻击(攻击 11.2)也是一个完美的交错攻击。同样, Lowe 对于 Needham-Schroeder 公钥认证协议的攻击(攻击 2.3)也是交错攻击。

通常, 存在双方认证缺陷的协议都可能受到交错攻击。

### 11.7.6 归因于类型缺陷攻击

在归因于类型缺陷攻击中, Malice 所利用的是我们在 11.3 节作出的关于诚实主体的一个约定: 诚实主体不具备把消息或者部分消息同该消息的语义联系起来的能力(见 11.3 节中注释 11.1 和例 11.1)。

典型的类型缺陷攻击包括欺骗某个主体, 使得他把一次性随机数、时戳或者身份等嵌入到某个密钥中去。如果协议的设计很糟糕, 协议中的消息部分的类型信息不明确, 就可能造成错误的理解。下面我们以 Neuman 和 Stubblebine[216]提出的协议为例来说明类型缺陷攻击[287, 70]。首先, 给出协议:



1. Alice  $\rightarrow$  Bob:  $A, N_A$ ;
2. Bob  $\rightarrow$  Trent:  $B, \{A, N_A, T_B\}_{K_{BT}}, N_B$ ;
3. Trent  $\rightarrow$  Alice:  $\{B, N_A, K_{AB}, T_B\}_{K_{AT}}, \{A, K_{AB}, T_B\}_{K_{BT}}, N_B$ ;
4. Alice  $\rightarrow$  Bob:  $\{A, K_{AB}, T_B\}_{K_{BT}}, \{N_B\}_{K_{AB}}$ 。

该协议的目的是在 Trent 提供的可信服务的帮助下, Alice 和 Bob 实现双方认证和认证的密钥建立。如果协议中一次性随机数和密钥都是随机的, 并有相同的规模, 那么该协议会受到类型缺陷攻击:

1. Malice (“Alice”)  $\rightarrow$  Bob:  $A, N_A$ ;
2. Bob  $\rightarrow$  Malice (“Trent”):  $B, \{A, N_A, T_B\}_{K_{BT}}, N_B$ ;
3. 跳过;
4. Malice (“Alice”)  $\rightarrow$  Bob:  $\{A, N_A, T_B\}_{K_{BT}}, \{N_B\}_{N_A}$ 。

该攻击中, Malice 使用一次性随机数  $N_A$  替换要建立的会话密钥  $K_{AB}$ , 因而 Bob 如果不能辨别类型的差别, 就会受骗而接受  $N_A$ 。实际上, 没有更好的机制可以防止 Bob 受到这样的欺骗。

类型缺陷通常因实现而定, 如果协议的描述对于协议中出现的变量没有提供足够明确的信息, 那么类型缺陷在实现中出现就是很平常的。Boyd[55]以 Otway-Rees 认证协议[228]为例说明了该问题, 在文[55]中, Boyd 讨论了在密码协议中避免隐含假设的重要性。

### 11.7.7 归因于姓名遗漏攻击

在认证协议中, 一般与某条个消息相关的名字可以从该消息上下文的其他部分或者从使用的加密密钥推断出来。然而, 当这些信息不能推断得出时, 姓名遗漏就是很大的错误, 会造成严重的后果。

似乎业内的专家们(密码学、计算机安全和协议设计领域的著名作者)更容易犯姓名遗漏错误, 这可能是因为这些专家希望设计出精巧的协议, 它应当包含很少的冗余信息。我们在 11.6.2 节和 11.6.3 节中所介绍的对 STS 协议两个版本两个攻击就是生动的例子。这里, 我们再给出一个例子。

Denning 和 Sacco 提出了一个公钥协议[95]作为他们对 Needham-Schroeder 对称密钥认证协议修改的另一种形式。该协议如下:

1. Alice  $\rightarrow$  Trent:  $A, B$ ;
2. Trent  $\rightarrow$  Alice:  $\text{Cert}_A, \text{Cert}_B$ ;
3. Alice  $\rightarrow$  Bob:  $\text{Cert}_A, \text{Cert}_B, \{\text{sig}_A(K_{AB}, T_A)\}_{K_B}$ 。

对该协议第三条消息加密是为了获得保密性和认证性。在 Bob 接收到 Alice 的消息后, Bob 会认为只有他和 Alice 知道  $K_{AB}$ , 这是因为加密部分是用 Bob 的公钥加密的, 并且 Bob 解密后可以验证 Alice 的签名。

遗憾的是, 该协议并不能保证 Alice 和 Bob 独享会话密钥  $K_{AB}$ 。Abadi 和 Needham 发现了一个简单而令人震惊的攻击[1]。该攻击中, Bob 把接收到的消息 3 发给另外一个主体 Charlie, Bob 就可以成功地欺骗 Charlie 相信 Charlie 跟 Alice “独享”会话密钥  $K_{AB}$ 。

3'. Bob("Alice") $\rightarrow$ Charlie: Cert<sub>A</sub>, Cert<sub>C</sub>, {sig<sub>A</sub>(K<sub>AB</sub>, T<sub>A</sub>)}<sub>K<sub>C</sub></sub>。

Charlie 会按照协议规定,相信该消息来自 Alice,并进而使用 K<sub>AB</sub> 来加密与 Alice 进行通信的机密消息。然而,此时 Bob 却可以阅读这些消息!

消息 3 本来的含义是:“在时刻 T<sub>A</sub>, Alice 说 K<sub>AB</sub> 是用于 Alice 和 Bob 通信的安全密钥”。该含义直接用协议来表达应该是:

3. Alice $\rightarrow$ Bob: Cert<sub>A</sub>, Cert<sub>B</sub>, {sig<sub>A</sub>(A, B, K<sub>AB</sub>, T<sub>A</sub>)}<sub>K<sub>B</sub></sub>。

在认证协议中,把参与者的身份明确表示,特别是在加密操作的范围内把身份明确表示,对于协议的设计者而言是“常识性”要求。然而,我们已经见证了富有经验的协议设计者违反“常识”的例子并不少。Abadi 和 Needham 已经对该“常识”做了论证,并把它作为认证协议设计的谨慎原则之一[1]。这里,我们再一次把该原则列出:

如果主体的身份信息对于该消息的含义是至关重要的,那么在消息中明确包含该主体的名字应当谨慎对待。

我们再次强调该谨慎原则并不是多余的:在 12.2 节我们会看到用于因特网安全的 IKE 协议[137]的当前版本中存在的姓名遗漏错误,而该协议当前版本是由经验丰富的计算机安全专家委员会经过多年的协议研发得到的。

### 11.7.8 密码服务滥用攻击

最后,我们介绍一种常见的协议设计缺陷:密码服务滥用。

密码服务滥用是指协议中的密码算法没有提供正确的保护,从而在协议中缺少所需的密码保护。这种缺陷会引发各种攻击。以下是最常见的两种:

- i) 归因于缺失数据完整性保护的攻击。我们将针对一个存在缺陷的协议给出一个攻击实例以说明数据完整性保护的重要性。在第 14 章,将给出更多用于攻击公钥密码体制的这类攻击的实例,并且我们也将介绍适应性主动攻击下安全的概念。在 17.2 节,我们研究认证协议分析的形式化方法时,会对这种协议缺陷做深入的研究。
- ii) 归因于缺失“语义安全”保护造成的机密性失败。在这种协议(密码体制)失败的类型中,Malice 可以从加密消息的密文中得到部分信息,从而在“完全或无”的保密意义下不必完全破解加密算法实现攻击目的(见 8.2 节的性质 8.2)。在第 14 章,我们研究语义安全的概念时,会给出许多这种类型的攻击。在第 14 章和第 15 章,我们会给出具有语义安全的密码技术。

在有关认证协议的文献中,这两种密码服务的滥用经常出现。显然,这表明那些协议的设计者并没有意识到“教科书密码”普遍存在的危险。

接下来我们介绍完整性服务缺失造成的缺陷。下面存在缺陷的协议是 Otway-Rees 协议的一个变形[228]。根据[62]的介绍,我们导出了该协议的变形。这个变形如协议 11.8 所示。

协议 11.8 使用了相当标准的在线认证服务器(Trent)技术,实现两个用户主体间的双方认证和认证的密钥建立。下面我们从 Bob 的角度来阐述协议的运行(从 Alice 的角度是类似的)。Bob 能够断定第 3 步接收的会话密钥是新鲜的,因为该密钥和 Bob 产生的一次性随机数进行

了密码综合。Bob 也能够断定他与 Alice 共享了这个会话密钥,因为此次运行的标识符和 Alice、Bob 的身份进行了密码综合,并且该综合由 Bob 本人创建,并由 Trent 验证。

### 协议 11.8 Otway-Rees 协议的修饰变型

前提: Alice 和 Trent 共享密钥  $K_{AT}$ ;

Bob 和 Trent 共享密钥  $K_{BT}$ ;

目标: Alice 和 Bob 相互认证并建立一个新的共享会话密钥  $K_{AB}$ 。

1. Alice  $\rightarrow$  Bob:  $M, Alice, Bob, \{N_A, M, Alice, Bob\}_{K_{AT}}$ ;
2. Bob  $\rightarrow$  Trent:  $M, Alice, Bob, \{N_A, M, Alice, Bob\}_{K_{AT}}, \{N_B\}_{K_{BT}}, \{M, Alice, Bob\}_{K_{BT}}$ ;
3. Trent  $\rightarrow$  Bob:  $M, \{N_A, K_{AB}\}_{K_{AT}}, \{N_B, K_{AB}\}_{K_{BT}}$ ;
4. Bob  $\rightarrow$  Alice:  $M, \{N_A, K_{AB}\}_{K_{AT}}$ 。

( \*  $M$  是协议运行的标识符,用于 Alice 和 Bob 跟踪他们之间的运行 \* )

该变形协议与原始的 Otway-Rees 协议的差别很小:在变形中的第 2 步,该差别体现在消息 2 中,Bob 加密的消息是分在两个密文分组中的,一个是对 Bob 的一次性随机数  $N_B$  加密,另一个是对消息的其他部分加密。而在原始的 Otway-Rees 协议中,Bob 的一次性随机数和消息的其他部分是被加密在同一密文组中(更精确地说,是被协议规定在同一密文组中): $\{N_B, M, Alice, Bob\}_{K_{BT}}$ 。

这里需要指出的是,在某些实现中,上面所说的变形可能根本称不上是一种变形:因为在实现中,不管协议规定使用一组还是两组,对一个长消息的加密总是用多个组实现的。这一点很重要,在我们关于这类协议失败原因讨论的最后,将再次阐述这一点。

与[62]中建议的修改相比,这个微小的变形实际上是原始协议修改中的一种做了更小改进的版本。在[62]中,认为 Bob 的随机数没有必要保密,因而 Bob 可以以明文形式直接发送。实际上,如果新鲜性标识符在消息 2 中是明文发送的,Bob 仍然能够用在消息 3 中返回的  $N_B$  来验证会话密钥  $K_{AB}$  的新鲜性。然而,为了更清楚地表述我们的观点,我们仍然在消息 2 中对新鲜性标识符加密。

协议 11.8 是存在缺陷的。攻击 11.7 表明了这一点,该攻击由 Boyd 和 Mao 发现[56]。

在该攻击中,Malice 首先假冒 Alice 发起和 Bob 的对话(步骤 1')。Malice 截获 Bob 发送给 Trent 的消息(步骤 2);然后把消息中 Alice 的身份改成 Malice 自己的身份,不改变 Bob 的第一个加密组(Malice 没有必要知道 Bob 的一次性随机数),用旧的密文组  $\{M, Malice, Bob\}_{K_{BT}}$  代替消息中 Bob 的第二个密文组  $\{M, Alice, Bob\}_{K_{BT}}$ ,其中旧的密文分组可以从 Malice 以前和 Bob 进行的某次合法通信的记录中获得。完成上述修改之后,Malice 冒充 Bob 把消息 2' 发送给 Trent (步骤 2')。此后 Trent 和 Bob 都不会发现有任何异常:Trent 认为申请认证服务的是他的两个客户 Malice 和 Bob,而 Bob 认为他正在同 Alice 运行协议。最后,Bob 会使用建立的会话密钥进行通信。由于 Bob 认为该会话密钥是他和 Alice 共享的,而实际是和 Malice 共享的,所以由 Bob 发送给 Alice 的机密信息都会被 Malice 得到。

## 攻击 11.7 对 Otway-Rees 协议修饰变形版本的攻击

前提:除协议 11.8 中规定的之外, Malice 和 Trent 共享密钥  $K_{MT}$ 。

( \* 所以 Malice 也是系统的正常用户 \* )

1. Malice(“Alice”)→Bob:  $M, Alice, Bob, \{N_M, M, Malice, Bob\}_{K_{MT}}$ ;
  2. Bob→Malice(“Trent”):  $M, Alice, Bob, \{N_M, M, Malice, Bob\}_{K_{MT}}, \{N_B\}_{K_{BT}}, \{M, Alice, Bob\}_{K_{BT}}$ ;
  - 2'. Malice(“Bob”)→Trent:  $M, Malice, Bob, \{N_M, M, Malice, Bob\}_{K_{MT}}, \{N_B\}_{K_{BT}}, \{M, Malice, Bob\}_{K_{BT}}$ ;
- ( \* 其中  $\{M, Malice, Bob\}_{K_{BT}}$  是旧的密文分组, 是 Malice 从先前同 Bob 的某次正常运行中保留下来的 \* )
3. Trent→Bob:  $M, \{N_M, K_{MB}\}_{K_{MT}}, \{N_B, K_{MB}\}_{K_{BT}}$ ;
  4. Bob→Malice(“Alice”):  $M, \{N_M, K_{MB}\}_{K_{MT}}$ 。

后果: Bob 认为他和 Alice 刚刚进行了一次对话并且和 Alice 共享了一个会话密钥。然而实际上, Bob 是和 Malice 进行了一次对话, 并且和 Malice 共享了那个会话密钥。

该攻击揭示了重要的一点: 用机密性服务来保护新鲜性标识符  $N_B$  就是在提供不正确的密码服务! 正确的服务是数据完整性服务, 必须用数据完整性来综合一次性随机数和参与主体的身份。如果能够提供合适的数据完整性保护,  $N_B$  确实可以明文发送; 而没有数据完整性服务, 对  $N_B$  加密也无济于事。

我们已经提到, 对于某些实现, 协议 11.8 并不能看做是原始 Otway-Rees 协议的变形。事实上确实是这样, 因为长消息的加密通常用多个组来实现。如果在实现当中这些组不是彼此密码综合的, 那么变形协议和原始协议所实现的代码相同。因此一个协议不能看做是另一个协议的变形。

分组密码常用的和标准的实现中, 一连串彼此分离的密文组采用密码技术相互链接在一起。最可能采用的是密码分组链接模式(CBC, 见 7.8.2 节)。我们应该注意, 在 CBC 模式中, 用密码技术链接的密码组实际上并没有受到数据完整性服务的保护, 这与通常的和误信的观点是相反的。因为没有数据完整性保护, 某些组可能会被修改而在解密时不会被发现。我们将在 17.2.1.2 节给出 CBC 不能提供数据完整性保护的原因。

## 攻击方法难以穷尽

还可以进一步列举认证协议的其他一些攻击方式, 例如“边信道攻击”(在 12.5.4 节中我们将看到这类攻击, 用于攻击 TLS/SSL 协议, 在那种情况下, 边信道攻击就是“定时分析攻击”)、“实现相关攻击”、“绑定攻击”和“封装攻击”(见[78]的第 4 节)或者“服务器误信攻击”(见[200]的 12.9.1 节)等。因为这些攻击类型与我们列出的某些攻击类型有交叉, 且即使把这些攻击全部包含, 我们仍然不能穷尽所有的攻击类型, 因此我们的列举到此为止。

即使业内专家非常小心,认证协议依然容易包含安全缺陷,这一事实促使研究者们寻求系统的方法来设计和分析认证协议。在第 17 章,我们会介绍设计和分析认证协议的形式化方法中的几个论题。

## 11.8 文献简记

认证协议是密码协议中一个很大的论题。我们推荐该领域的几个重要的参考文献:

- Burrows、Abadi 和 Needham 所著的 *A logic of authentication* [62]。该文是一篇基本文献。许多安全协议的论文都引用该文。从中可以找到许多早期的认证协议以及这些协议早期暴露出来的一些安全缺陷。
- Moore 所著的 *A survey on various ways cryptographic protocols fails* [206, 207]。该文是一篇重要文献。该文对各种不同形式的密码协议失败原因做了很好的介绍,这些失败不是由密码算法本身的任何弱点引起的,而是由于使用密码算法的方式造成的,所使用方式要求某个密码算法提供该算法本身不能提供的密码服务。
- Abadi 和 Needham 所总结的 *Prudent engineering practice for cryptographic protocols* [1]。该文列出了 11 个启发式的原则,用于指导协议设计者设计出好的协议。这些原则构成了协议发展的工程细则,成为协议设计者的菜单:“我是否已验过这类攻击?”。这是一篇优秀论文,并将证明对于协议设计者是相当有用的。
- Clark 和 Jacob 所著的 *A survey of authentication protocol literature* [78]。该文包含大量的认证协议和认证的密钥建立协议,其中许多协议都附有对它们的攻击。该文还包含广泛并附有很好注释的文献概述。该文是协议研发者们的一个基本文献。作为 Clark 和 Jacob 工作的进一步发展,人们建立了一个称为“安全协议开放知识库(SPORE)”的 Web 站点。SPORE 的 Web 地址是 <http://www.lsv.ens-cachan.fr/spore/>。
- 一本即将面世的由 Boyd 和 Mathuria 所著的书,名为 *Protocols for Key Establishment and Authentication* (信息安全和密码学丛书,出版社:Springer, ISBN:3-540-43107-1)。该书是第一本全面论述认证的密钥建立协议的书。书中认证的密钥建立协议包括使用对称和非对称密码技术的基本认证协议,包括基于群、会议密钥以及基于口令的认证协议等,作者对它们进行了全面的描述,并尽量详细地公布、描述和解释了已知的缺陷。该书也使得理论工作者和实际开发者都可以很快地根据他们的需求查询到相关协议,并知道现有的在文献中已被攻破的协议。除了对协议统一清晰地描述之外,该书还包括了对所有的主要攻击类型的描述,并把大部分协议按照其属性和资源需求进行了分类。该书也包含适合于研究生学习的教学内容。

## 11.9 本章小结

本章对认证这一领域进行了深入而广泛的讨论。包括基本概念(数据源、实体、认证的密钥建立、单方、双方、活现性),好的认证协议结构(国际标准所推荐的),标准协议,以及几个有趣并且有用的协议(例如,一次性口令、EKE、STS)和攻击类型的分类。

在密码协议领域中,作为一个学术上活跃的研究论题,认证协议是很重要但还远未成熟的课题。本章并没有全面地覆盖这一论题。因此我们为有兴趣将此课题作为学术研究方向的读



者扼要地列出了几个参考文献。对这些读者而言,本书后面有一章(第 17 章,认证协议的形式化分析方法)也是进一步的学习材料。

认证协议在现实应用中是很重要的。本章已初步涉及了几个应用方面。下一章我们将介绍认证协议在实际中的应用。

## 习题

- 11.1 描述以下安全服务的差别:数据完整性、消息认证、实体认证。
- 11.2 何谓新鲜性标识符?
- 11.3 某个主体所执行的新鲜密码操作是否就意味该主体所发送消息的新鲜性?
- 11.4 Alice 对某个密文(例如,用 AES-CBC 加密得到)解密后,看到了一个有效的新鲜性标识符(例如,Alice 发送的一次性随机数),那么 Alice 能否断定该密文消息是新鲜的?
- 11.5 为什么某个协议加密消息的数据完整性对于该消息的机密性是重要的?
- 11.6 在 11.4 节,我们介绍了最基本的认证协议结构。在这些结构中,标准结构和非标准结构的本质区别是什么?
- 11.7 辨别 Woo-Lam 协议(协议 11.2)所采用的某个非标准结构。  
提示:观察 Bob 和 Trent 消息 4 和消息 5 的交互中所用的安全服务,并与(11.4.7)中的结构进行比较。
- 11.8 下列三个攻击哪一个是常见的?(i)Wiener 对有缺陷的 ISO 协议版本的攻击(攻击 11.1),(ii)对“惟认证”STS 协议的“替换证书签名攻击”(攻击 11.2),(iii)Lowe 对于 Needham-Schroeder 公钥认证协议的攻击(攻击 2.3)。
- 11.9 在计算机中,每一个 ASCII 码都由 8 比特表示。那为什么由 8 个 ASCII 字符组成的口令所包含的信息量少于 64 比特?
- 11.10 在基于口令的认证协议中,何谓加盐操作?其作用又是什么?
- 11.11 在 UNIX 操作系统的口令认证协议(参见 11.5.1 节和协议 11.3)中,密码变换  $f(P_U)$  使用了 DES 加密函数。那么该协议是否使用了 DES 解密函数?讨论该变换与非标准认证机制中此类变换的主要差别?
- 11.12 S/KEY 协议(协议 11.4)在本质上使用的是和 UNIX 口令认证协议(协议 11.3)相同的密码变换。那么为什么 S/KEY 协议是有缺陷的而后者却没有缺陷?
- 11.13 EKE 协议(协议 11.5)使用了非对称密码技术,那么该协议是否是基于公钥的认证协议呢?
- 11.14 我们指出了“惟认证”STS 协议存在缺陷(攻击 11.2)。请修补该协议,消除其中的缺陷。
- 11.15 在 11.6.3 节,我们论述了对验证者的身份签名可以修补 STS 协议中存在的小缺陷(攻击 11.3 所展示的缺陷)。然而,这样的修补会损害协议的匿名性(可否认性)。请给出一个不同的修补,它不包含对身份签字。  
提示:双方实际上没有把共享会话密钥和意定的通信方绑定;这是我们认为协商的密钥没有双方认证的原因,见 11.6.1 节我们对于该协议设计属性的讨论。



## 第 12 章 认证协议——实践篇

### 12.1 引言

前一章关于认证协议的讨论是从学术观点出发的:我们研究了好的(和标准的)认证协议结构,从文献中引入了一些重要的认证协议和认证技术,并系统地考察了对认证协议的各种“学院派攻击”。但是,我们在应用方面涉及得不多。毋庸置疑,在实际应用中,认证协议必定存在需要解决的许多实际问题,其中一些问题是颇具挑战性的。

本章我们集中讨论实际应用中的认证问题。我们将介绍并讨论一些已经提出将用于或者已经广泛地用于实际中的各种重要的认证协议。本章介绍的协议都是事实标准或者工业标准。

我们首先介绍现实世界中的因特网密钥交换协议(IKE)[137,160]。该协议是因特网工程任务组(IETF)标准(IPsec)中的认证机制。该协议(协议族)在通信底层的网络层运行,包含认证协议和认证的密钥交换协议。通过进一步学习,我们会了解网络层通信是如何发生的,进而就会明白第 11 章中 Malice 的各种攻击是如何通过修改网络层协议处理的地址信息获得成功的,并将看到在网络层提供安全服务对于抵御各种攻击是非常有效的。我们还将看到 IKE 中具有挑战性的一个问题,那就是如何使该协议族适合提供可选的隐私属性,该属性是为了使网络层通信不会损害高层应用提供的隐私服务而设置的。

接下来,我们介绍安全壳(SSH)协议[306,309,310,307,308]。它是基于公钥的认证协议族,适合于从不可信任终端远程安全地接入计算机资源(安全远程登录)。该协议是开放系统环境下安全远程登录的事实标准,已经在全球范围内广泛地使用。SSH 是一种客户-服务器协议。其服务器端运行的操作系统主要是 UNIX<sup>①</sup>或者其通用版本 Linux;在客户端,运行的操作系统则包括 Windows 等。该协议面临的具有挑战性的问题在于以某种和谐的方式提供安全服务:不安全的系统已经在广泛地使用,应该以最小的中断在这个早已使用的不安全系统中添加安全的解决方案(这本质上是后向兼容问题)。

然后,我们将介绍另一个重要的并且已经广泛使用的认证协议:Kerberos 认证协议[204,170]。该协议是另一个常用操作系统 Windows 2000 操作系统的网络认证基础,这个操作系统广泛地用于企业环境,在这个环境中的用户有权使用企业网络范围内提供的各种服务,但是不能持有使用不同服务的多个密码证书(让单个用户记住许多口令是不现实的,为单个用户维护多个智能卡也是不经济的)。我们将会看到 Kerberos“单点登录”认证架构应用在这种环境下很合适。

最后,我们简述安全套接层协议(SSL)[138],或者按照国际上工业标准团体 IETF 的命名称为传输层安全(TLS)协议。在本书写作时,该协议已经可以称为最广泛使用的基于公钥的认证技术。现在,该协议已经是每一个 Web 服务器和浏览器集成软件的一部分。通常,使用该

---

<sup>①</sup> UNIX 是贝尔实验室的商标。

协议只是为了完成单方认证(服务器向客户认证),这是该认证协议在典型的客户-服务器设置下的应用。尽管该协议背后的思想非常简单(这是本章将介绍的 4 个用于实际应用的协议中最简单的一个),该协议实际实现时也是复杂的。从这里,我们可以看到任何简单认证协议实际实现都不是一件简单的工作。

### 12.1.1 章节概述

12.2 节介绍 IPSec 和 IKE 协议。12.3 节介绍 SSH 协议。在 12.4 节,首先介绍适合使用 Windows 2000 操作系统的企业单点登录场景,然后描述该系统网络认证的基础——Kerberos 协议。最后,在 12.5 节综述 SSL(TLS)协议。

## 12.2 用于因特网的认证协议

前面已经介绍了各种密码技术,这些技术用于保护通过开放网络传输的消息。到目前为止,本书中介绍的技术都在高通信层或者说在应用层提供保护。在应用层提供保护意味着所保护的对象仅限于消息的内容部分,而消息的地址信息属于低层信息,所以没有提供保护。

然而,对于因特网的安全通信,在低层提供保护并且使保护对象包含地址信息和内容时,才能够提供非常有效的保护。这是因为,如我们在 11.7 节所见,修改消息的地址信息是 Malice 得以发起各种攻击的主要方法。

本节,我们首先介绍低层通信协议是如何处理消息的。在这一部分,我们将明白 Malice 如何利用没有安全保护的通信协议来完成其攻击。然后,我们学习由国际标准组织为因特网安全提出的一套认证协议。这套协议总称为**因特网密钥交换(IKE)**协议,目的在于使用我们第 11 章介绍的认证技术来保护低层通信协议的消息。我们将分析 IKE 中几个重要的“模式”,并揭露其存在的一些缺陷。最后报导研究界对于 IKE 的一些批评意见和关注的问题。

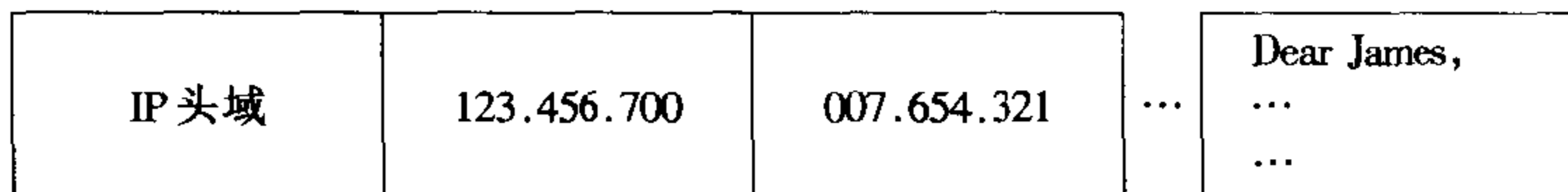
### 12.2.1 IP 层通信

因特网是由称为“节点”的计算机和设备组成的巨大开放网络。每个节点都被分配有全局惟一的某个网络地址,从而发往该节点或者从该节点发出的消息都携带该网络地址。运用网络地址来处理消息传输的协议称为**因特网协议**(简称 IP),这样,某个节点全局惟一的网络地址就称为该节点的 IP 地址。根据 ISO“开放系统互联(ISO-OSI)七层参考模型”(例如[229]的 416-417 页或[159]的 1.5.1 节),IP 在“第 3 层”工作(也称为网络层或 IP 层)。许多通信协议,其中包括终端用户调用的许多认证协议,工作在“第 7 层”(也称为应用层)。这也是我们称 IP 为“低层”通信协议而称其他协议为“高层”通信协议的另一个原因。

IP 层通信是以“IP 数据包”的形式实现的。图 12.1 描述的是某个没有密码保护的 IP 数据包,该数据包的前三个域有明显的含义,第四个域“上层域”有两层含义:(i)描述随后的高层协议,并处理该 IP 数据包(例如,“传输控制协议”TCP);(ii)IP 数据包所承载的数据。

下面以电子邮件为例来说明使用 IP 数据包进行因特网通信的过程。首先考虑不安全的情况,也就是 IP 数据包没有密码保护的情况。假设两个电子邮件地址为 James\_Bond @ 007.654.321 和 Miss \_ Moneyppenny @ 123.456.700。这里,James \_ Bond 和 Miss \_

Money Penny 是用户名, 又称为“终端身份”; 007.654.321 和 123.456.700 是两个 IP 地址<sup>①</sup>, 例如, 前者是某个多用途掌上设备的 IP 地址, 后者是某个办公室计算机的 IP 地址。从 IP 层来看, 从 Miss\_Money Penny @ 123.456.700 发送给 James\_Bond @ 007.654.321 的电子邮件和下图类似:



为了便于说明, 我们只给出了在“上层域”中数据域的内容, 略去了协议处理的信息(该例中, 略去的协议处理信息为“SMTP”, 表示简单邮件传送协议[166])。注意这两个终端身份会在某个“IP 头域”中出现, 所以当 James\_Bond 接收到电子邮件后, 他就知道是谁发送的。

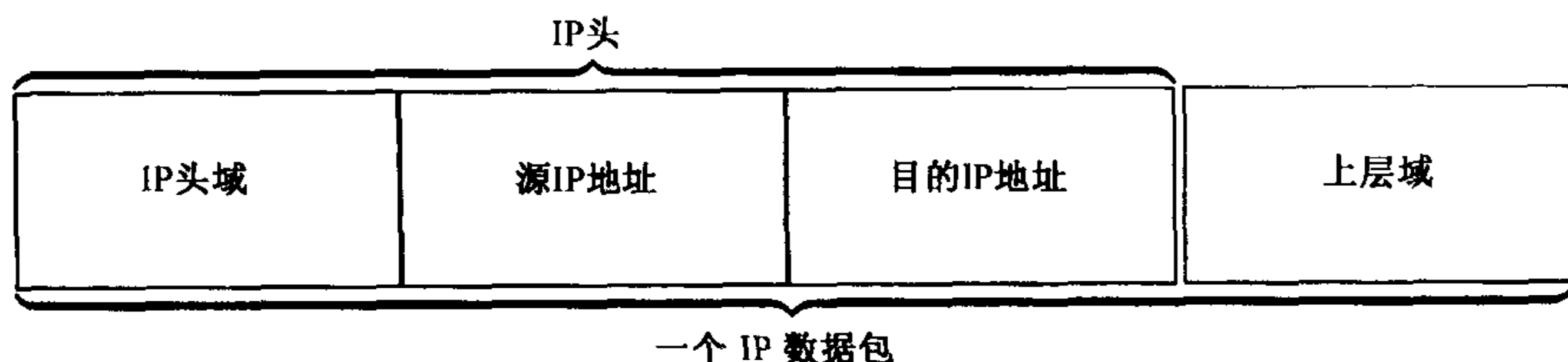


图 12.1 未加保护的 IP 数据包

通信双方可能希望通过使用共享密钥或者公钥完成端到端加密以达到机密通信的目的。因为端到端加密是在应用层处理的, 因而“IP 数据包”中只有第四个方框中的内容是加密的。如果他们使用的 IP 协议不提供安全性, 那么“IP 头”中的数据就得不到保护。通过修改这些数据, Malice 就可以完成我们在 11.7 节中列出的大部分攻击。下面介绍 Malice 是如何完成攻击的。

### 12.2.2 IP 安全协议 (IPSec)

因特网工程任务组(IETF)制定了用于 IP 安全的系列标准, 这些标准统称为 IPSec [165, 163]。简单地说, IPSec 就是给 IP 数据包中前三个方框构成的“IP 头”增加密码保护(见图 12.1)。IPSec 规定了用于“IP 头”的强制性认证保护和可选的机密性保护, 后者用于保护出现在“IP 头域”中的终端身份信息。

我们应该注意到, 由于 IP 层缺少安全机制, 没有保护传输的“IP 头”, 才使得 Malice 可以对因特网通信发起各种攻击, 例如欺骗(冒充)、窃听(截听)和会话劫持(对合法通信会话的欺骗和窃听的混合攻击)等。举例来说, 如果 Malice 截取了从 James\_Bond @ 007.654.321 发出的 IP 数据包, 然后把“源 IP 地址”域数据拷贝到“目的地址”域, 然后再发送出去, 那么该数据包就会回到 James\_Bond @ 007.654.321。如果该修改是不可检测的, 那么 Malice 的这一行为就构成了 11.7.4 节所介绍的“反射攻击”。进一步, 如果 Malice 伪造了所截获数据包的“源地址”域内容和终端身份信息(比方说 Miss\_Money Penny), 那么 James\_Bond @ 007.654.321 就会被欺骗相信该消息来自于修改后的源地址。这正是我们在应用层已给出的攻击情形, 即

Malice(“Miss\_Money Penny”) → James\_Bond: ...

<sup>①</sup> 通常为了便于记忆, IP 地址会映射为某个“域名”。例如, 007.654.321 可能会被映射为“域名”spy1.mi.five.gb。



实际上,我们在 11.7 节列出的各种攻击都需要 Malice 对“IP 头”中 IP 地址和终端身份进行某种程度的修改。因此,在 IP 层提供安全保护能够非常有效地防止各种攻击,因为这种保护能够检测所有对“IP 头”信息的修改。通常来说,在 IP 层提供的安全服务可以对所有高层应用提供广泛的保护。

另外,对于**防火墙**<sup>①</sup>之间的业务流,因为防火墙这样的节点会屏蔽许多在它“里面”或者说在它“后面”的节点,所以 IP 层保护会使得该防火墙“里面”任意节点的 IP 地址被加密。这意味着未经授权通过防火墙这样的行为可以通过密码方法来防止,这是一种很强的保护方式。如果 IP 层没有安全保护,防火墙技术只能使用很弱的方式,即利用一些勉强称为“秘密”的信息来防止非法行为,例如 IP 地址、机器名和用户名等,这时未经授权通过防火墙会变得容易一些。人们一致认同在 IP 层提供安全服务是一个明智的举措。

### 12.2.2.1 IPSec 中的认证保护

因特网协议(IP)已经从第 4 版(IPv4)发展到第 6 版(IPv6)。IPv6 数据结构是由称为数据报的 32 比特分组构成的。IPv6 是带有 IPSec 保护的,IPv6 数据包有一个新的称为“认证头”(AH)的域(比较图 12.2 和图 12.1)。在 IP 数据包中,AH 在“IP 头”和“上层域”之间。AH 的长度是可变的,但必须是 32 比特数据报长度的倍数。AH 域细分为几个子域,其中包含为 IP 数据包提供密码保护所需的数据。

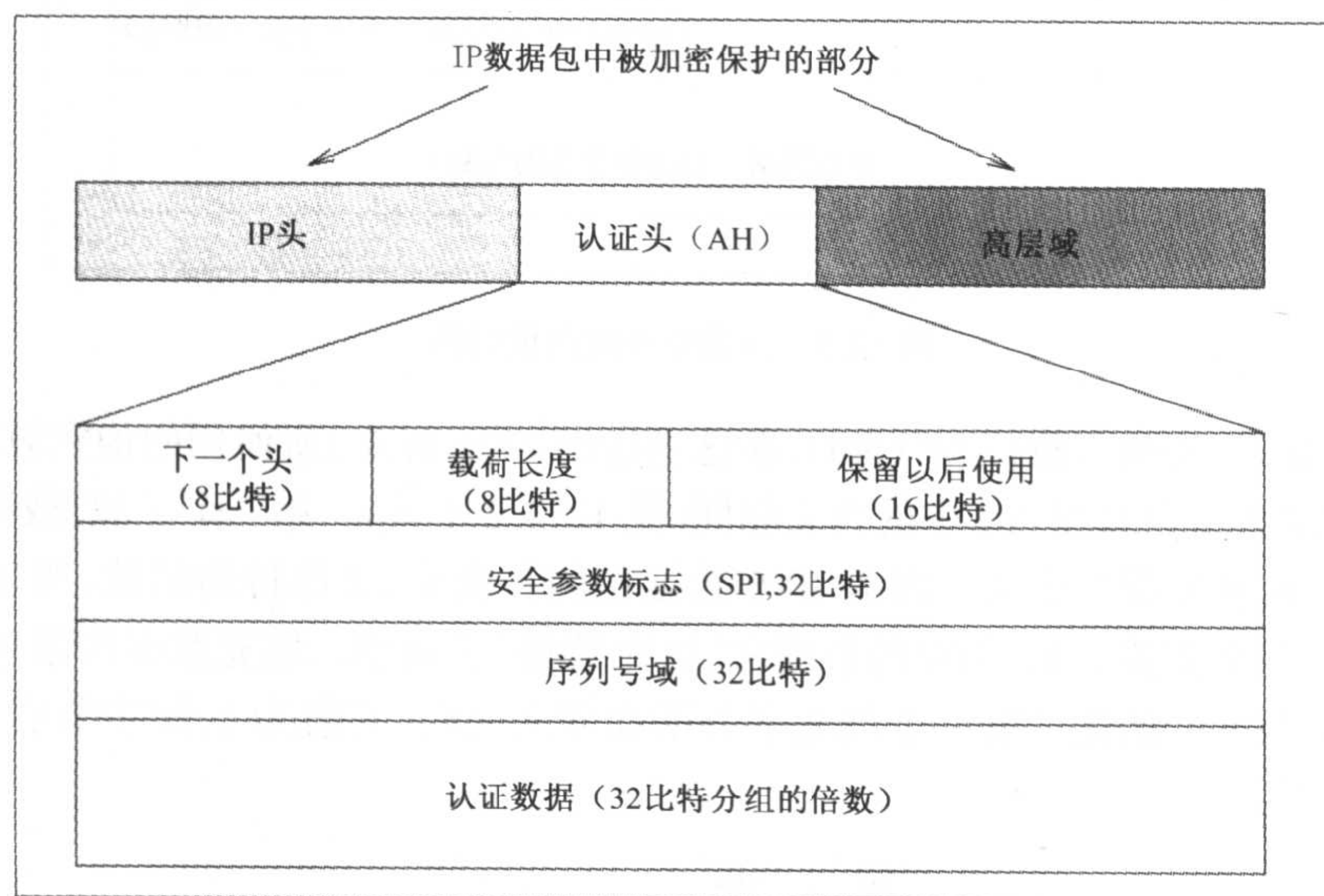


图 12.2 认证头的结构及其在 IP 数据包中的位置

认证(实际上是指带有源身份信息的数据完整性)是 IPSec 的强制性服务,提供该保护所需的数据包含在 AH 的两个子域中。其中一个子域称为“安全参数标志”(SPI),该子域包含长 32 比特的某个任意值,指出(惟一标志了)用于该 IP 数据包认证服务的密码算法。另一个子域称为“认证数据”,包含消息发送方为接收方生成的认证数据,用于接收方验证数据完整性(因

① 防火墙是某台有特定目的的计算机,该计算机把一系列受到保护的计算机和设备连接到因特网。这样,从因特网访问受保护的计算机和设备时需要知道某种身份信息和 IP 地址信息。



此这部分数据称为完整性校验值,ICV)。该 IP 数据包的接收方能够使用密钥和 SPI 惟一标志的算法重新生成“认证数据”,然后比较自己生成的认证数据和接收到的认证数据,完成 ICV 校验,其中使用的密钥将在 12.2.3 节讨论。

子域“序列号”能够抗击 IP 数据包重放。AH 第一个数据报的其他子域,包括“下一个头”、“载荷长度”和“保留以后使用”都没有安全方面的意义,因此这里不对它们进行说明。

#### 12.2.2.2 IPSec 中的机密性保护

机密性(加密)是 IPSec 的一个可选服务。为了获得机密性,一个称为“封装安全载荷”(ESP)的 32 比特倍长的数据报在 IP 数据包中占有一定的空间并且有详细的说明。ESP 可以位于图 12.2 中 AH 之后的第二个阴影区(“高层域”)。ESP 的格式如图 12.3 所示。

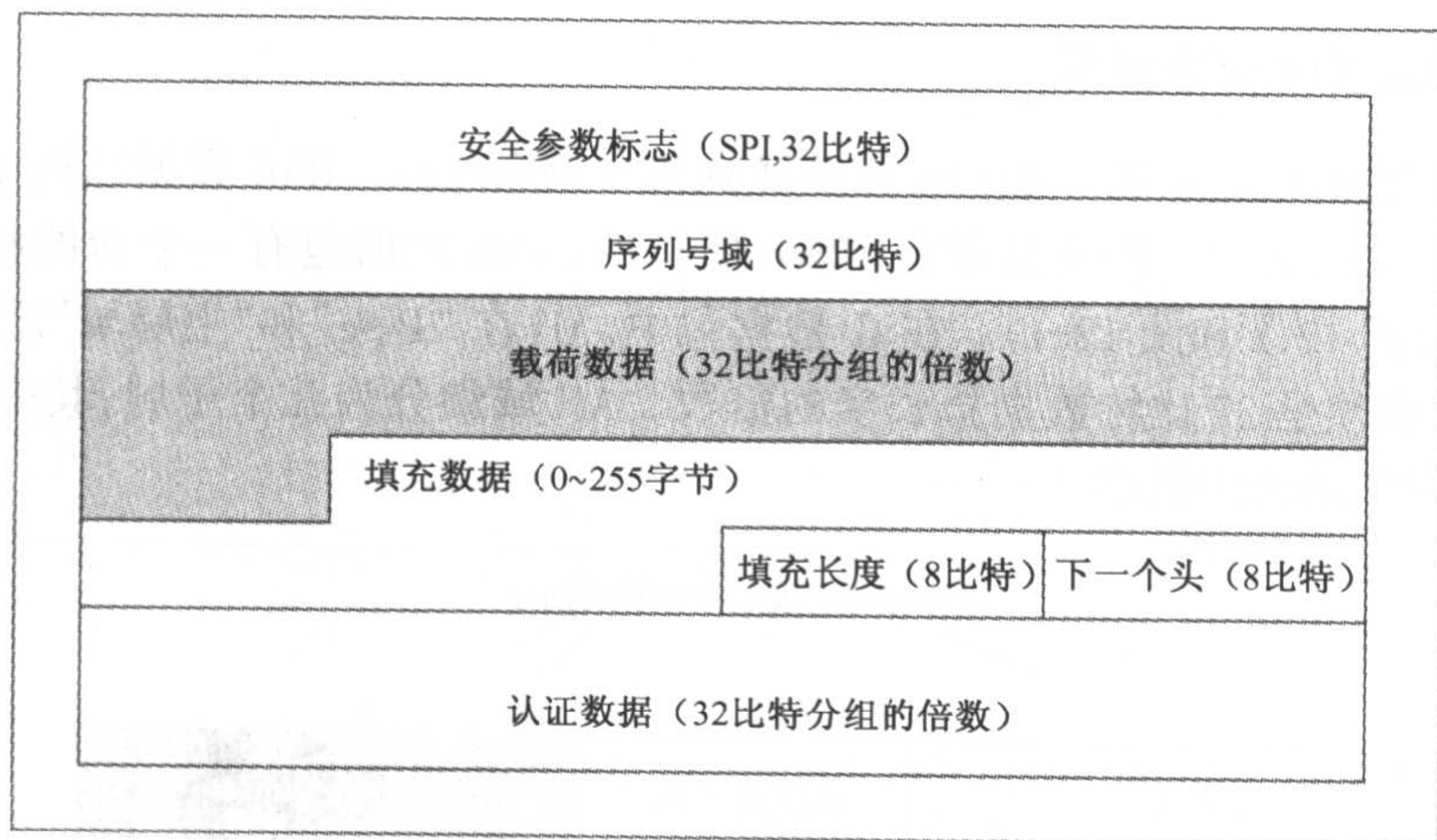


图 12.3 封装安全载荷的结构

第一个子域是“安全参数标志”(SPI),在这里指出(惟一标志)所使用的加密算法。第二个子域“序列号”的含义与其在 AH 中的含义相同(见 12.2.2.1 节)。第三个子域“载荷数据”长度可变,其内容为机密数据的密文。因为 IPv6 数据包的长度是 32 比特的倍数,所以变长的“载荷数据”对应的明文需要填充,填充的数据在“填充数据”子域中。填充数据用系列的(无符号的单字节)整数数据初始化。第一个填充字节附在明文之后,其值为 1,该字节后面的填充字节依次递增如下:

$$'01' \parallel '02' \parallel \dots \parallel 'xy'$$

其中,‘xy’是 16 进制的数值,‘01’ $\leq$ ‘xy’ $\leq$ ‘FF’。因此,填充字节的最大数目为‘FF’=255<sub>(10)</sub>。填充数据的字节长度称为“填充长度”。最后,“认证数据”子域的含义与其在 AH 中的含义相同。

读者应该注意到 ESP 中的“认证数据”和 AH 中这一部分数据的区别。ESP 中的认证数据是提供 ESP 数据包(即 ESP 数据中不包含“认证数据”子域的所有其他子域)的密文(即 ESP 数据包所有子域减去其“认证数据”子域)的数据完整性并且是可选的[164]。而在 AH 中,“认证数据”保护 IP 数据包的数据完整性并且是强制使用的。

在 ESP 中,“认证数据”是可选的,这一点实际上是错误的。我们将在 12.2.5 节对该错误进行讨论。



### 12.2.2.3 安全连接

IPSec 的中心概念之一是“安全连接”(SA)。SA 用下面的数组惟一标志：

(SPI, “IP 目的地址”, “服务标志符”)

其中“服务标志符”指 ESP 或认证。

本质上, IPSec 可以认为是 AH + ESP。当两个网络节点在 IPSec 保护下通信时, 它们必须协商一个 SA(用于认证)或者协商两个 SA(用于认证和加密), 并协商这两个节点间所共享的会话密钥以便它们能够执行密码操作。该协商运用我们接下来要介绍的因特网密钥交换协议来完成。

### 12.2.3 因特网密钥交换(IKE)协议

在增加了“认证头”(AH)和“安全封装载荷”(ESP)后, IPSec 就具备了对 IP 数据包实行密码保护的能力。然而, 为了使这种能力起作用, 通信双方必须首先协商 SA(包括密钥、密码算法、密码参数)。该协商由因特网密钥交换(IKE)协议 [137, 160] 实现。IKE 是用于 IPSec 认证密钥交换协议的 IETF 当前标准。

IKE 是认证和认证密钥交换协议族的总称。该协议族中每个协议都是由三部分合成的, 分别是“Oakley”(Oakley 密钥确定协议 [227])、“SKEME”(通用因特网安全密钥交换机制 [174])和“ISAKMP”(因特网安全连接和密钥交换协议 [189])。

Oakley 描述了系列的密钥交换方式, 其中每一种方式都称为一种模式并且 Oakley 对每种模式所提供的安全服务都给出了详细的描述(这些安全服务包括会话密钥的完善前向保密、隐藏终端身份、双方认证等)。SKEME 描述了认证的密钥交换技术, 该技术支持通信各方对于曾经存在连接的可拒绝特性(指否认发生在通信双方间的连接, 它基于共享密钥, 是 IKE 和 IKE v2 的一个特性, 在后面会专门对其进行讨论)和快速密钥更新。ISAKMP 为通信双方达到认证和认证的密钥交换提供了通用的框架。该框架用于协商和确认各种安全属性、加密算法、安全参数、认证机制等。这些协商结果统称为“安全连接”(SA)。然而, ISAKMP 没有提供任何具体的密钥交换技术, 所以它可以支持各种密钥交换技术。

作为以上工作的合成体, 可以认为 IKE 是适用于双方参与的协议族。它们具有认证的会话密钥交换的特性, 大部分是用 Diffie-Hellman 密钥交换机制完成的, 包含许多的可选项用于参与双方以在线的方式协商和确认选择项。

IKE 协议包括两个阶段, 分别称为“阶段 1”和“阶段 2”。

阶段 1 假设参与密钥交换的双方中的每一方都拥有对方预先知道的某个身份信息, 并且所拥有的该身份信息与某种密码能力相关, 这种密码能力可以发送给对方用于验证。该能力在对称加密体制下可以通过预共享密钥获得, 在公钥密码体制下, 该能力通过私钥获得, 该私钥与某个公钥的可信拷贝匹配。阶段 1 通过向通信对方发送证明其密码能力的消息, 试图建立双方认证<sup>①</sup>, 建立用于阶段 1 的会话密钥, 必要时, 该会话密钥可用于保护阶段 2 的交换; 或者进一步作为 IKE 各阶段交换的输出, 用于其他高层应用程序进行安全通信。

<sup>①</sup> 接下来我们就会看到, IKE 阶段 1 的一些模式并没有达到双方认证, 因为某个实体可以被完全欺骗去相信他和意定的通信方共享会话密钥, 而实际上他和另一方共享那个会话密钥。



通信的两个实体完成阶段 1 交换之后,同一对实体可以运行多个阶段 2 的交换。阶段 2 通常称为“快速模式”。这一阶段依赖于阶段 1 所建立的会话密钥。运行多个阶段 2 交换是因为这样能够让用户建立起多个具有不同安全属性的连接,比如分别用于“仅数据完整性”、“仅机密性”、“用短密钥加密”、“用强密钥加密”等。

为了更好地研究 IKE,下面我们集中介绍几个 IKE 阶段 1 的运行模式。

#### 12.2.3.1 IKE 阶段 1

IKE 阶段 1 有八种变形,这是因为存在三种密钥类型(预共享对称密钥、公钥加密密钥、公钥验证签名密钥)。此外,基于公钥加密密钥又有两个协议版本,其中一个可用于代替另一个,然而为了后向兼容,文档中也必须包含前者。因此,实质上就存在四种密钥类型(预共享对称密钥、老式公钥加密密钥、新式公钥加密密钥、公钥验证签名密钥)。对于每种密钥类型,存在两种密钥交换模式:主模式和进取模式。

每种主模式都有 6 次消息交换:3 次是从发起者(简称  $I$ )到响应者(简称  $R$ ),3 次是从  $R$  到  $I$ 。IKE 中主模式是强制实现的,也就是说,用户不能在没有运行主模式的条件下首先运行进取模式。

每种进取模式都只有 3 次消息交换: $I$  发起对话,发送一条消息, $R$  响应一条消息,然后  $I$  发送最后一条消息结束对话。进取模式是可选的,也就是说,这种模式可以忽略。

对于 IKE 阶段 1,我们将只描述和分析“基于签名的模式”。其他的模式通常采用对新鲜性标志符进行加密-解密的机制来获得认证。我们已经(见 11.4.1.5 节)把这种机制称为非标准机制;在 17.2 节,我们将进一步对其进行批评讨论。

#### 12.2.3.2 基于签名的 IKE 阶段 1 主模式

基于签名的 IKE 阶段 1 主模式(也称为“用签名认证”,见[137]的 5.1 节)如协议 12.1 所示。该协议是在多个协议的影响下产生的,但从根本上可以追溯到两个协议:STS 协议(协议 11.6)和 Krawczyk 提出的 SIGMA 协议[173](SIGMA 协议将在 12.2.4 节讨论)。

第一对消息交换是  $I$  发送  $HDR_I$  和  $SA_I$  到  $R$ ,  $R$  响应  $HDR_R$ ,  $SA_R$ 。头消息  $HDR_I$  和  $HDR_R$  中分别包含“小甜饼” $C_I$  和  $C_R$ 。 $C_I$  用于  $R$  维护  $I$  的会话状态信息, $C_R$  亦然。两个安全连接中, $SA_I$  详细列出了  $I$  所建议使用的安全属性, $SA_R$  详细列出了  $R$  所选择的某一种安全属性。

第二对消息交换包含了 Diffie-Hellman 密钥交换所需的素材。

消息 5 和消息 6 中所用的加密算法、签名算法、计算消息杂凑值以用于签名的伪随机函数等,都是在 SA 协商中确定的。

基于签名的 IKE 阶段 1 主模式与 STS 协议(协议 11.6)有些类似,然而存在以下两个主要区别:

- i) STS 协议没有对证书加密,而该模式中证书被加密。我们介绍 STS 协议时已经指出证书加密使得匿名性成为可能(见 11.6.1 节)。这对于防火墙后面的  $I$  和(或)  $R$  来说是个有用的特性,也是可以实现的。
- ii) STS 协议中的签名没有包含会话密钥,然而该模式中签名的消息和会话密钥  $g^{xy}$  同时作为伪随机函数  $\text{prf}$  的输入,其中会话密钥作为种子。这样该模式中的签名就只对于拥有会话密钥的双方来说是可验证的。

## 协议 12.1 基于签名的 IKE 阶段 1 主模式

1.  $I \rightarrow R: \text{HDR}_I, \text{SA}_I;$
2.  $R \rightarrow I: \text{HDR}_R, \text{SA}_R;$
3.  $I \rightarrow R: \text{HDR}_I, g^x, N_I;$
4.  $R \rightarrow I: \text{HDR}_R, g^y, N_R;$
5.  $I \rightarrow R: \text{HDR}_I, \{\text{ID}_I, \text{Cert}_I, \text{Sig}_I\}_{g^{xy}};$
6.  $R \rightarrow I: \text{HDR}_R, \{\text{ID}_R, \text{Cert}_R, \text{Sig}_R\}_{g^{xy}}.$

符号(\* 为了便于说明,我们省略了一些无关紧要的细节。省略的内容不会影响该协议的功能,特别地,不会影响我们稍后所描述的攻击\*)

$I, R$ : 分别代表发起者和响应者。

$\text{HDR}_I, \text{HDR}_R$ : 分别代表  $I$  和  $R$  的消息头,其中分别包含  $C_I$  和  $C_R$ ,也就是  $I$  和  $R$  的“小甜饼”<sup>①</sup>,分别为这两个实体维护会话的状态信息。

$\text{SA}_I, \text{SA}_R$ : 分别代表  $I$  和  $R$  的安全连接。两个实体使用  $\text{SA}_I$  和  $\text{SA}_R$  来协商该协议的当前运行中将使用的安全参数;协商的参数包括:加密算法、签名算法、获得消息的杂凑值以用于签名的伪随机函数等。 $I$  可以提出多种选项,而  $R$  必须用其中的某个选项来回应。

$g^x, g^y$ : 分别代表  $I$  和  $R$  的 Diffie-Hellman 密钥交换素材。

$\text{ID}_I, \text{ID}_R$ : 分别代表  $I$  和  $R$  的终端身份。

$N_I, N_R$ : 分别代表  $I$  和  $R$  的一次性随机数。

$\text{Sig}_I, \text{Sig}_R$ : 分别代表  $I$  和  $R$  所创建的签名。签名消息分别为  $M_I$  和  $M_R$ 。其中,

$$M_I = \text{prf}_1(\text{prf}_2(N_I | N_R | g^{xy}) | g^x | g^y | C_I | C_R | \text{SA}_I | \text{ID}_I)$$

$$M_R = \text{prf}_1(\text{prf}_2(N_I | N_R | g^{xy}) | g^y | g^x | C_R | C_I | \text{SA}_R | \text{ID}_R)$$

其中,  $\text{prf}_1$  和  $\text{prf}_2$  是在  $\text{SA}$  协商中确定的伪随机函数。

## 12.2.3.3 基于签名的 IKE 阶段 1 主模式的认证失败

与 STS 协议的情形类似,这种 IKE 运行模式中签名消息只包含了签名者的终端身份信息,不包括意定通信方的终端身份信息。缺少这种明确的信息使得该模式也存在与 STS 协议类似的认证失败缺陷,从而不能抵御 Lowe 攻击(攻击 11.3)。攻击 12.1 说明了这种缺陷。Meadows 也证明了 IKE 的此种运行模式存在类似缺陷[197]。

利用该缺陷, Malice 可以成功地欺骗  $R$  相信  $I$  发起并且完成了与  $R$  的一次对话。而实际上  $I$  并没有发起并完成同  $R$  的对话。注意到  $R$  被完全欺骗,这表现在以下两个方面:首先,  $R$  错误地接受了一个通信方,并认为与这个错误接受的通信方共享某个会话密钥;其次,没有人会告诉  $R$  存在任何异常。所以攻击 12.1 确实展示了一种认证失败。

① “小甜饼”是一种纯文本的字符串,存在于远端主机系统的内存或者某个文件中,目的是维护客户-服务器通信会话的状态信息。

### 攻击 12.1 基于签名的 IKE 阶段 1 主模式的认证失败

( \* Malice 对  $I$  使用他的真实身份信息,而对  $R$  则冒充  $I$  \* )

1.  $I \rightarrow \text{Malice} : \text{HDR}_I, \text{SA}_I;$
- 1'  $\text{Malice}("I") \rightarrow R : \text{HDR}_I, \text{SA}_I;$
- 2'  $R \rightarrow \text{Malice}("I") : \text{HDR}_R, \text{SA}_R;$
2.  $\text{Malice} \rightarrow I : \text{HDR}_R, \text{SA}_R;$
3.  $I \rightarrow \text{Malice} : \text{HDR}_I, g^x, N_I;$
- 3'  $\text{Malice}("I") \rightarrow R : \text{HDR}_I, g^x, N_I;$
- 4'  $R \rightarrow \text{Malice}("I") : \text{HDR}_R, g^y, N_R;$
4.  $\text{Malice} \rightarrow I : \text{HDR}_R, g^y, N_R;$
5.  $I \rightarrow \text{Malice} : \text{HDR}_I, \{ \text{ID}_I, \text{Cert}_I, \text{Sig}_I \}_{g^{xy}};$
- 5'  $\text{Malice}("I") \rightarrow R : \text{HDR}_I, \{ \text{ID}_I, \text{Cert}_I, \text{Sig}_I \}_{g^{xy}};$
- 6'  $R \rightarrow \text{Malice}("I") : \text{HDR}_R, \{ \text{ID}_R, \text{Cert}_R, \text{Sig}_R \}_{g^{xy}};$
6. Dropped.

后果:  $R$  认为刚才和他对话并和他共享会话密钥的是  $I$ , 而  $I$  却认为刚才是和 Malice 进行了一次不成功的通信。  $R$  根本不会被通知存在任何异常, 并且也许会拒绝来自  $I$  的服务; 实际上  $R$  进入了这样一种状态, 只接受来自  $I$  的服务请求(也许直到“超时”后,  $R$  才会脱离这样的状态)。

这种认证-失败攻击也有很好的理由被称为“拒绝服务攻击”。在 IKE 中, 经过一次成功的阶段 1 交换,  $R$  作为服务器会保留和  $I$  的当前状态信息, 以便在进一步进行多个阶段 2 交换通信时继续使用确认的会话密钥。然而, 经过攻击 12.1 中所展示的一种攻击实例以后,  $I$  不会继续同  $R$  进行任何对话, 而此时  $R$  却在维护  $I$  的状态信息, 分配和  $I$  通信所需的资源, 并等待  $I$  接下来的对话。如果 Malice 是以一种分布的方式展开此攻击的, 例如伙同其队友在因特网上同时对某个服务器展开攻击, 那么受到攻击的服务器服务其他诚实用户的能力会急剧下降, 甚至不能提供任何服务。并且注意到该攻击对于 Malice 及其合谋者来说并不需要精妙的技术和复杂的计算, 因而这种分布式拒绝服务攻击会非常有效。

该攻击之所以奏效是因为协议的签名消息中只包含签名者的身份信息, 所以可以用这样的一条消息来欺骗不是签名者意定通信方的某个第三方主体。如果签名消息中包含双方意定主体的终端身份, 那么该消息就只能用于这两个主体, 不能用于其他任何目的。

我们再一次看到了姓名遗漏缺陷所引起的攻击的普遍性。

#### 12.2.3.4 基于签名的 IKE 阶段 1 进取模式

基于签名的 IKE 阶段 1 进取模式是相应主模式版本的简化版: 进取模式中没有使用加密, 它只有 3 次消息交换而不是 6 次。基于签名的进取模式如下所示, 其符号与主模式(协议 12.1)相同:

1.  $I \rightarrow R: \text{HDR}_I, \text{SA}_I, g^x, N_I, \text{ID}_I$
2.  $R \rightarrow I: \text{HDR}_R, \text{SA}_R, g^y, N_R, \text{ID}_R, \text{Cert}_R, \text{Sig}_R$
3.  $I \rightarrow R: \text{HDR}_R, \text{Cert}_I, \text{Sig}_I$

从表面上看,该模式与“惟认证 STS 协议”(协议 11.7)很类似,因为两者都从原来的协议中去掉了加密操作。但仔细观察会发现其中的区别:“惟认证 STS 协议”中的签名消息不涉及会话密钥,而在该模式中,签名消息和会话密钥  $g^y$  同时作为伪随机函数 prf 的输入,其中会话密钥为伪随机函数的种子。所以在该模式中,只有持有会话密钥的主体才能够验证签名。这种差别使得“替换证书签名攻击”(攻击 11.2)不能用于这种运行模式的攻击。

然而,由于另外的原因,该模式依旧不能达到双方认证。一种类似的“拒绝服务攻击”能应用在这种模式上,实质上就是 Lowe 对于 STS 协议的攻击(见攻击 11.3)。只是在对该模式的攻击中, $I$  被完全欺骗, $I$  相信他和  $R$  运行了某次对话并且共享会话密钥,而  $R$  却不这么认为。我们将把该攻击的具体实现作为习题留给读者(见习题 12.6)。

进一步应该注意到,如果该模式使用的签名机制具有消息恢复的特性,那么 Malice 可以获得更多的好处。例如,从给定的签名消息中, Malice 能恢复出  $\text{prf}_2(N_I \parallel N_R \parallel g^y)$ , 这样 Malice 就能够使用他自己的身份信息和证书以及恢复出的消息重新构造一个签名。这样 Malice 就能够发起“替换证书签名攻击”,在攻击 11.2 中我们已经见过这种对“惟认证 STS 协议”的攻击。这次攻击可以说是完美的攻击,因为在 Malice 的设计下, $I$  和  $R$  之间交错的运行实例都会成功,而两个诚实主体  $I$  和  $R$  不会发现任何异常。注意到某些签名机制确实有消息恢复的特性(例如[222],它甚至已经标准化[152])。因此,通信双方协商的签名机制可能具有消息恢复特性。在 12.2.5 节,我们会讨论 IKE 支持各种灵活选项这一特性所带来的问题。

由于没有使用加密或者 MAC, IKE 进取模式不可能具有“看似合理的拒绝”这一特性,上述特性我们将在 12.2.4 节讨论。在不需要这一特性时,此种认证-失败缺陷的修正有标准的方法:在两个签名消息中分别包含两个意定主体的终端身份信息,于是签名消息只能用于这两个意定主体的此种交换模式中,而在任何其他场景中都不能使用。

我们将在 12.2.4 节讨论修补认证缺陷并同时保持可否定性的一些方法。

### 12.2.3.5 对 IPSec 和 IKE 的其他安全分析

几位研究者曾对 IKE 进行了安全分析工作。

Meadows 使用 NRL 协议分析器(自动穷举缺陷检测器,17.5.2 节介绍[196,195])已经发现快速模式(IKE 阶段 2 交换)容易受到反射攻击[197]。

Ferguson 和 Schneier 从密码角度对 IPSec 进行了全面的评价[109]。

Bellovin 分析了 IPSec 中的一个严重问题:某个 IPSec 运行模式中的某个选项使得密文消息在数据完整性方面得不到保护[28]。通过攻击实例我们已经意识到且现已知道以下事实:没有完整性保护的加密完全达不到机密性要求(见 11.7.8 节)。在后面的几章(第 14 章~第 17 章)中我们会进一步看到,大部分加密算法,如果其输出的密文在数据完整性方面没有得到保护,就不能提供合适的机密性保护。然而,这个危险的选项似乎还没有得到 IPSec 标准制定团体的注意(见下文),这可能是由于 IPSec 规范中过高的系统复杂性。

### 12.2.4 IKE 中看似合理的可否认性

在本书写作时, IKE 第 2 版(IKEv2)规范已经发布[160]。IKEv2 把“阶段 1 交换”的多种“模式”合成为单一的 IKEv2“阶段 1 交换”。然而, 当前规范[160]限制了协议使用数字签名作为认证的基础(见[160]的 5.8 节)。Boyd、Mao 和 Paterson 分析表明 IKEv2“阶段 1 交换”本质上存在着 IKE 第 1 版中攻击 12.1 所利用的同一弱点[57]。

IKEv2 中采用了一种可选项, 这一特征被称为通信的“看似合理的可否认性”[141], 做这种否认的实体可能曾卷入和某个通信伙伴进行的连接。这一特性来源于 Krawczyk 提出的 SIGMA 协议结构(SIGMA 是“Sign”和“MAC”的合成, 见[173]中的解释)和 Canetti 与 Krawczyk[68], 它允许某个实体“看似有理地”否认曾和某个通信方建立过连接。其所以希望在 IP 层提供这种否认连接特性, 是因为它允许在较高层以可靠的质量提供各种奇妙的隐私服务, 比如匿名性。在 IP 层所造成的隐私损害能够在应用层引起无法弥补的隐私损害。例如, 假设某个身份信息和某个 IP 地址对应, 如果不具备拒绝特性, 那么某个奇妙的密码协议在高层运行时所提供的匿名性就必然会变得无效。

SIGMA 设计方案中“看似合理的可否认性”可以用下面两行消息描述。在协议 12.1 中, 这两行消息应该位于协议的第 5 行和第 6 行:

$$I \rightarrow R: s, ID_I, \text{Sig}_I("1", s, g^x, g^y), \text{MAC}(g^{xy}, "1", s, ID_I)$$

$$R \rightarrow I: s, ID_R, \text{Sig}_R("0", s, g^y, g^x), \text{MAC}(g^{xy}, "0", s, ID_R)$$

这里( $s$  是会话标志符)双方都可以验证对方的签名, 然后使用共享的会话密钥验证对方的 MAC, 并因此确认另一端是否为意定的通信方。这时, 如果通信双方都删除会话密钥, 他们谁也不能向第三方证明他们之间确实发生过此次对话。

容易看出, 上述结构中依旧包含认证-失败缺陷, 该缺陷在攻击 12.1 中已经给出了示例。Canetti 和 Krawczyk 确实曾预料到存在另一种不大重要的攻击, 该攻击中 Malice 只是简单地使  $I$  收不到  $R$  发给他/她的最后一条消息。他们建议通过在最后增加一条从  $I$  到  $R$  的回执消息来防止这种“截去最后消息的攻击”(见[68]的注释 2)。这样, 最后一条消息由  $R$  (通常是作为服务器)接收,  $R$  就能够检测到这种攻击, 并在发现攻击后, 重新设置状态信息, 释放资源。所以, 该协议不容易受到拒绝服务攻击。实际上, 最后增加的这条回执消息还有另外的作用: 它可以消除认证-失败缺陷(取决于回执消息所使用的密码表述)。然而很明显, 这样修正协议的方法在实际中是不理想的, 因为这种方式包括了额外的流量, 增加了协议的复杂性。

因为可否认特性是有用的, 所以我们应该在修补认证-失败缺陷时保留这一属性。鉴于这一点, 我们建议改进 SIGMA 的设计方案, 使之成为下列两行:

$$I \rightarrow R: s, ID_I, \text{Sig}_I("1", s, g^x, g^y), \text{MAC}(g^{xy}, "1", s, ID_I, ID_R)$$

$$R \rightarrow I: s, ID_R, \text{Sig}_R("0", s, g^y, g^x), \text{MAC}(g^{xy}, "0", s, ID_R, ID_I)$$

这样, 通信双方依旧能够非显式地签署他们的身份, 因而保留了“看似合理的不可否认性”。然而, 他们必须明确验证在 MAC 中通信对方的身份。

注意这种设计中的否认连接属性并不是完全合格的, 因为如果某一方(称之为“叛徒”)保留了会话密钥  $g^{xy}$ , 那么他就仍然能够在某次协议运行完以后, 拿出证据向第三方证明某个指定的(认证的)实体曾经涉及此次连接。这显然是可能的, 因为叛徒只要完全按照协议运行时



的验证过程重新来一遍就可以证明这一点。这就是我们在可否认属性前加上“看似有理”这样一个修饰语的原因。

在 13.3.5 节,我们将介绍一种新的实际可用的密码原型,它可以提供一种绝对意义下的可否认属性的认证服务。

### 12.2.5 对 IPSec 和 IKE 的批评意见

对 IPSec 和 IKE 最主要的批评在于它们所强化的系统的复杂性和缺乏清晰性。IPSec 和 IKE 中包含太多的选项和太多的灵活性。对于相同的或者类似的事情常常有多种做法。Kaufman 对于 IKE 中密码协商的次数有一个计算:1 次“必须”,806 399 次“可以”[159]。高系统复杂性引起的直接后果是系统规范的极其晦涩。这种晦涩并不是一件好事:这会很容易地使分析该系统的专家产生困惑,并进而妨碍这些专家看到系统安全弱点,或者误导系统实现人员,使他们编出存在缺陷的实现程序。

Ferguson 和 Schneier 认为这种高系统复杂性是一种典型的“委员会效应”[109]。他们认为“委员会总是在不停地增加特性、选择和额外的灵活性来满足委员会内各派系的需要,这是人所共知的”。的确,如果委员会效应,例如增加系统复杂性,对于普通(功能性)标准是严重损害的(如我们有时所经历的),那么对于安全标准就是灾难性的。

其实,高度的灵活性和大量的可选择项所引起的严重问题,并不仅是使分析该系统的专家在理解其表现时极度困难,也不仅是使得实现人员编码后很难得到运行正确的系统,而且还在于该系统中的某些选项可能本身就是危险的。在 12.2.3.4 节,我们描述了一种可能的情况:通信双方选择的签名机制具有消息恢复的属性,Malice 可以对 IKE 基于签名的进取模式发起成功的交错攻击。下面,我们看一下关于这种危险性的另一个例子。

关于上述危险性的这个例子源于一篇解释性论文的摘录,该论文的标题是“理解 IPSec 协议族”[12]。该文发表于 2000 年 3 月,提供了对 IPSec 和 IKE 各种级别的解释,从网络安全的一般概念到 IPSec 和 IKE 的各种具体属性。下面的摘录(源于[12]的第 6 页)是对可选项“封装安全载荷(ESP)中的认证”的解释(ESP 是密文分组,其加密对象是在 IP 数据包中传输的一些机密数据,见 12.2.2.2 节):

ESP 认证域是 ESP 中的一个可选项,包含完整性校验值(ICV)——本质上是对 ESP 剩余部分(减去认证域本身)所计算的数字签名。ESP 认证域长度可变,这取决于使用的认证算法。如果不选择 ESP 认证服务,可完全省略该域。

在该解释中,我们看到存在一个选项可以省略对密文全部的数据完整性保护。在 11.7.8 节中我们已经看到,并且在后续的章节会继续看到,没有数据完整性(该摘录中的“认证”)保护的加密通常是危险的,对于大多数加密算法,如果没有数据完整性保护,就不能提供合适的机密性保护。在 IPSec 中的这个安全问题,Bellare 在 1996 年就已经指出并做了批评(见 12.2.3.5 节最后一段),然而在 4 年后,这一点仍然保留了下来并且被解释为该系统的特性(IPSec 解释这篇论文发表于 2000 年 3 月)!我们认为正是 IPSec 规范高度的复杂性才导致这种危险的错误被隐藏。

Aiello 等[10]批评 IKE 在通信和计算上的高(系统设计)复杂性。他们认为 IKE 中的协议容易受到拒绝服务攻击:Malice 及其合伙人只要在因特网上初始化许多连接请求就可以做到



这一点,因为此时服务器将不得不维护大量有状态的“小甜饼”。Aiello 等提出了一个称为 JFK (Just Fast Keying)的协议,并建议 JFK 作为 IKE 的替代者。Blaze 透露了他们把协议称为 JFK 的原因[40]:

我们决定取的这个名字 Ike 曾是以美国为中心的一个的双关语,它是美国总统艾森豪威尔的昵名,艾森豪威尔曾有句口号是“我喜欢 Ike”。我们不喜欢 Ike,所以我们愿意看到 Ike 的后继者。我们的协议称为 JFK,是“Just Fast Keying”的缩写,然而它同时也是在以后一段时间内成为艾森豪威尔的继任总统肯尼迪(John Fitzgerald Kennedy)首字母的缩写。我们希望永远不在达拉斯讨论这个协议。如果在达拉斯还会再次举行 IETF 会议的话<sup>①</sup>,我们决不会在那儿提我们的协议。

## 12.3 安全壳(SSH)远程登录协议

安全壳(SSH)[306,309,310,307,308]是一套基于公钥的认证协议族。使用该协议,用户可以通过不安全网络,从客户端计算机安全地登录到远端的服务器主机计算机,并且能够在远端主机安全地执行用户的命令,能够在两个主机间安全地传输文件。该协议是工业界的事实标准,在运行 UNIX 和 Linux 操作系统的服务器计算机上应用广泛。该协议的客户端可以在任何操作系统平台上运行。该协议主要在 UNIX(Linux)服务器上运行的原因是这些操作系统具有开放架构,支持远端用户交互的命令会话。

SSH 协议的基本思想是客户端计算机用户下载远程服务器的某个公钥,然后使用该公钥和用户的某些密码证件建立客户端和服务端之间的安全信道。现在假设用户的密码证件是用户的口令,那么该口令就可以用服务器的公钥加密,然后发送给服务器。这与我们前面章节看到的简单口令认证协议相比,在安全上已经有了很大的进步。

### 12.3.1 SSH 架构

SSH 协议的运行环境是两台互不信任的计算机和连接它们的不安全通信网络。其中一台计算机称为远程服务器(主机),另一台称为客户端,用户使用 SSH 协议从客户端登录到服务器。

SSH 协议族主要包含三部分:

- SSH 传输层协议[310]提供服务器向用户的认证。该协议基于公钥,协议的前提(也就是预先输入到该协议的条件)是服务器端拥有一对称为“主机密钥”的公钥,在客户端拥有公开的主机密钥。该协议的输出是服务器到客户端的单方认证安全信道(在机密性和数据完整性方面安全)。典型情况下,该协议在 TCP(传输控制协议)和 IP(因特网协议)连接上运行,但也可以在其他任何可靠的数据流连接上使用。
- SSH 用户认证协议[307]。该协议运行在 SSH 传输层协议建立的单方认证信道上。该协议支持使用各种单方认证协议来达到从客户端用户到服务器的实体认证。为了使这一方向的认证成为可能,远程服务器端必须预先知道用户密码证件的相关知识,也就是说,用户必须是服务器能够识别的用户。这一部分使用的认证协议可以基于公钥,也可

<sup>①</sup> 第 34 次 IETF 会议于 1995 年 12 月在得克萨斯州的东北部城市达拉斯举行。

以基于口令。例如,可以使用基于口令的简单认证协议(协议 11.3)。这一部分使用的某个协议运行后的输出和第一部分协议运行后的输出,共同构建了在服务器端和客户端某个用户之间的双方认证安全信道。

- SSH 连接协议[308]。该协议运行在前面两个协议建立的双方认证安全信道上。这一部分实现了具体的安全加密信道,并将其隧道化为几个安全逻辑信道,使其能够在更广范围的安全通信用途上使用。这一部分用标准的方法提供交互的会话。

很明显,SSH 连接协议不是认证协议,不在本书的讨论范围之内。而 SSH 用户认证协议族可以看做是一种统称,即各种应用在该协议族中的标准(单方)认证协议的统称,而在第 11 章我们对此已经做了介绍(仍然需要注意在 12.3.4 节将要讨论的一个要点)。因此,我们需要介绍的只是 SSH 传输层协议。

### 12.3.2 SSH 传输层协议

在 SSH 的最新版本[309,310]中,SSH 传输层协议应用了 Diffie-Hellman 密钥交换协议,其中服务器要对其密钥交换素材签名,这样就达到了从服务器到客户端的单方认证。

#### 12.3.2.1 服务器的主机密钥对

每一台服务器主机都持有该主机的一对密钥——公钥和私钥。一个主机可以有多对主机密钥用于支持多种不同算法。如果服务器主机确实有密钥对,那么该主机必须至少拥有一对具备公约算法的密钥对。当前因特网-草案[309]规定,默认情况下的公钥算法是 DSS(数字签名标准,见 10.4.8.2 节)。用于当前版本([306],撰写此书的时候)的默认公钥算法是 RSA 签名(见 10.4.2 节)。

服务器主机密钥对(公钥、私钥)在密钥交换时使用:服务器使用私钥对交换的密钥素材签名;客户端使用服务器主机的公钥来验证客户是否真的在同正确的主机对话。为了使这一点成为可能,客户端必须预先知道关于服务器主机的公钥信息。

对于服务器主机公钥,SSH 支持两种不同的信任模式:

- 客户端有本地数据库,该数据库中含有关联服务器主机名和相应服务器主机公钥的信息。这种方法不需要集中管理的基础设施(称为公钥基础设施,将在第 13 章介绍),因而不需要可信第三方的协作。这种方法的缺点是保存关联信息(主机名、主机公钥)的数据库需要由用户来维护,这可能会成为用户的负担。我们会举例说明一个实际的方法(见 12.3.2.2 节),通过该方法,远程用户可以获得主机公钥的某个真实拷贝。
- 关联信息(用户名,主机公钥)的真实性由某些可信的证书发放机构(CA)来保证,此时使用的技术将在第 13 章介绍。客户端只需要知道 CA 的公钥,就能够验证所有该 CA 所证明的主机公钥的有效性。

第二种可供选择的方法简化了密钥管理问题,因为在理想情况下,客户只需要安全地保存一个 CA 的公钥就可以了(在这里安全是指数据完整性)。另一方面,每一个主机的公钥都必须在被某个 CA 正确地证明之后,才可能开始认证过程。同时,这种方法对集中的基础设施过于信任。

因为现在因特网上还没有广泛配置的公钥基础设施(PKI,第 13 章介绍),此时作为可选项,第一种信任模式使得该协议在过渡时期具有更大的可用性,这种过渡期会持续到 PKI 确实

出现。同时,这种信任模式与传统的方法(例如 UNIX 远程会话口令 `rlogin`、`rsh`、`rftp` 等)相比确实提供了更高级别的安全。

### 12.3.2.2 对服务器主机公钥认证的一种现实方法

用于用户获得服务器主机公钥真实拷贝的一种可行方法,是该用户随身携带含有该服务器主机公钥的一个拷贝,并且总是在密钥交换协议运行前才把该拷贝送入客户端计算机。例如在用户外出旅行时,该用户可以随身携带一个存储有服务器主机公钥的软磁盘。在当前 SSH 协议工作版本中[306],如果客户端运行的是 UNIX 或者 Linux 操作系统,那么客户端机器所用的服务器主机公钥就存储在文件 `$ Home/.ssh/known_hosts` 中。用户在外出旅行时应该在物理上保证服务器主机公钥的安全(例如,把公钥存储在用户外出旅游时随身携带的软磁盘中),这种安全是指数据完整性。当客户端计算机运行 Windows 操作系统时,服务器公钥可能只存在于客户端机器的内存中。这样,公钥可以从服务器上实时下载(当然是通过公开信道),同时下载的还有该公钥的“指纹”(见下一段),该“指纹”将提交给用户,以辨别公钥的真实性。

另一种方法,用于用户证实其从公开信道上所下载的服务器主机公钥的真实性,是使用电话进行声音认证。首先,服务器主机公钥通过公开信道由用户下载到客户端的机器中。然后该主机公钥 16 进制的“指纹”会显示给用户。该“指纹”如下:

$$\text{"fingerprint"}(\text{host key}) = H(\text{host key})$$

其中  $H$  是协商的密码杂凑函数,例如 SHA-1。对于 SHA-1,整个“指纹”的长度是 160 比特,因而可以通过电话以依次读 40 个 16 进制字符的形式传递。所以,用户可以给远程服务器站点的安全管理员打电话,验证“指纹”是否真实,方法就是比较安全管理员阅读的字符和客户端计算机计算得到的字符。这样,客户端用户和远程服务器端的安全管理员使用其声音证实了主机公钥的正确性。我们在这里假定用户和安全管理员可以互相辨认对方的声音。

这些方法并没有稳固的安全基础,但是在现实中在很大程度上是可操作的,并且具有实际安全性。目前在因特网 PKI 还未完全建立的情况下,这些方法是有用的。

### 12.3.2.3 密钥交换协议

密钥交换连接通常由客户端发起。服务器在某个特定的端口监听,等待客户端的连接请求。多个客户可能会连接到同一服务器计算机上。

SSH 的最新版本[309,310]应用 Diffie-Hellman 密钥交换协议(见 8.3 节),达到了建立会话密钥的目的。在描述该协议时,我们使用以下符号:

- $C$ : 客户端;
- $S$ : 服务器端;
- $p$ : 安全大素数;
- $g$ :  $GF(p)$  子群  $G_q$  的生成元;
- $q$ : 子群  $G_q$  的阶;
- $V_C, V_S$ :  $C$  和  $S$  各自的协议版本;
- $K_S$ :  $S$  的主机公钥;
- $I_C, I_S$ :  $C$  和  $S$  的“密钥交换初始消息”,表示这部分协议开始前要交换的消息。

密钥交换协议如下所示：

1.  $C$  生成随机数  $x(1 < x < q)$  并计算

$$e \leftarrow g^x \pmod{p}$$

$C$  将  $e$  发送给  $S$ ；

2.  $S$  生成随机数  $y(0 < y < q)$  并计算

$$f \leftarrow g^y \pmod{p}$$

$S$  接收  $e$ ；然后计算

$$K \leftarrow e^y \pmod{p}$$

$$H \leftarrow \text{hash}(V_C \parallel V_S \parallel I_C \parallel I_S \parallel K_S \parallel e \parallel f \parallel K)$$

$$s \leftarrow \text{Sig}_S(H)$$

$S$  将  $K_S \parallel f \parallel s$  发送给  $C$ ；

3.  $C$  验证  $K_S$  确实是  $S$  的主机密钥(使用合适的方法,例如证书、可信本地数据库或者在 12.3.2.2 节所描述的一种方法)；

$C$  计算

$$K \leftarrow f^x \pmod{p}$$

$$H \leftarrow \text{hash}(V_C \parallel V_S \parallel I_C \parallel I_S \parallel K_S \parallel e \parallel f \parallel K)$$

$C$  验证对  $H$  的签名  $s$ ；验证通过,  $C$  就接受本次密钥交换。

密钥交换完成之后,通信双方之间的通信内容可以使用确认的会话密钥  $K$  进行加密。下一步,双方执行的是 SSH 用户认证协议[307],可能使用的是任何一种已知的单方认证技术。完成用户认证之后,客户端的用户就可以使用 SSH 连接协议[308]请求服务了。

### 12.3.3 SSH 策略

SSH 协议的目标之一是以一种渐进的方式来增强因特网的安全。该协议允许客户端使用“各种合适的方式”(例如,在 12.3.2.2 节给出的方法)验证服务器公钥的真实性,这一点清楚地表明了 SSH 的策略包括快速配置和后向兼容。

目前,因特网的公钥基础设施在因特网范围内还不是完全可用,那么使用 SSH 后得到的改良安全未必特别安全,但已经相当安全,并且比不使用 SSH 好得多。易用和快速配置的解决方案是 SSH 很有价值的一个方面,也正因为这一点,SSH 成为通用的配置,广泛应用于 UNIX 和 Linux 平台。

从上面认证技术在实际系统中应用的实例,我们也看到公钥密码学对于获得简单的解决方案起着至关重要的作用。服务器主机公钥在不可信任环境中(例如客户端或者服务器到客户的中间路由环境)只需以公开的方式存在,这样对这个重要密钥素材的管理就变得极为简单。如果该协议基于对称密钥密码技术,问题将变得极为复杂。

### 12.3.4 警告

最后,我们给出忠告,用户应该妥善处理其在 SSH 用户认证协议中使用的密码证件。这

个密码证件,可以基于公钥,也可以基于口令或者基于安全硬件令牌环,它将在客户端计算机运行该协议时使用,而客户端通常被认为是属于不可信环境的。

在当前 SSH 协议的工作版本中[306],用户基于公钥的密码证件(例如,用户公钥所对应的私钥)被加密,加密密钥是客户端用户的口令,所得密文存储在客户端机器的一个文件中,文件名为 `$HOME/.ssh/identity`(在客户端计算机操作系统为 UNIX 或者 Linux 时)。在协议的客户端部分运行时将读取该文件,并要求用户输入口令。自然,这时用户应该能保证运行在客户端计算机上的协议是真实的。并且,为了减小离线攻击得到用户私钥的危险(其算法是输入用户的公钥,通过搜索用户的口令找出与该公钥匹配的用户私钥),用户应该在协议运行完后从客户端计算机上安全删除加密私钥的文件 `$HOME/.ssh/identity`。

基于安全硬件令牌环的机制应该算是保护用户密码证件最安全的方法。该机制中用户使用小巧的硬件令牌环,尺寸约一掌之宽或钥匙环大小。该令牌环有一个视窗,显示大量独立且较短的数据,这些数据因与服务器主机同步而不断变化,并且这些数据是绑定到某个单独用户的,绑定的方式是与服务主机共享口令。当然,因为口令规模很小,用户应该保证该令牌环的物理安全并在其丢失后立即上报。

## 12.4 Kerberos 协议及其在 Windows 2000 系统中的实现

假设 Alice 是某跨国公司的雇员,那么她可能面临(接受)各种信息资源和服务。例如, Alice 可以从“本地服务器”得到各种一般的计算机网络服务(如万维网、电子邮件等);而对于“项目服务器”, Alice 和她的团队则是惟一的合法用户群,同时是与她们工作相关的数据的所有者;在“人力资源服务器”上, Alice 会管理与 HR 相关的问题,例如调整她本人下月的薪水中用于购买公司股票的比例;如果 Alice 是经理,她可能需要更新其下属在 HR 数据库的表现记录;从“知识产权服务器”, Alice(发明者)可以对她的专利文件进行操作;使用“开支服务器”, Alice 可以在商务外出回来之后通过该服务器报销。类似的例子还有很多,不再一一列举。

在企业环境中,用户(员工或消费者)通常可以使用分布在企业范围内的各种信息服务,这些服务器通常由企业不同的部门维护。因此,各种信息服务器可以在不同的地理位置上工作(可能是全球范围)。按照网络组织来说,这些服务器分布在不同的网络域。为了安全地使用这些服务(上一段中我们列出的各种例子中都涉及严重的机密信息),用户需要出示各种密码证件以便通过对她/他的认证,获得服务授权。然而,要求用户维护多种不同的密码证件是不现实和不经济的,如在要求用户记忆各种口令或者持有多个智能卡等。

适合上述环境的一种合适的网络认证解决方案是 Kerberos 认证协议[204, 170]。其基本思想是使用可信第三方把某个用户引见给某个服务器,引见方法是在用户和服务器间分发会话密钥建立安全信道。该思想由 Needham 和 Schroeder[215]提出,体现在 Needham-Schroeder 认证协议中(协议 2.4)。因为该协议原始版本存在缺陷(见 2.6.4.2 节),所以 Kerberos 本质上使用的是 Needham-Schroeder 协议的带有时戳的版本。

接下来,假设协议 2.4 中 Alice 代表某个用户,她与可信第三方(协议中的 Trent)共享长期密钥;同时假设该协议中 Bob 代表某个服务器,他与可信第三方共享长期密钥。当 Alice 想要使用 Bob 提供的服务时,她就发起和 Trent 运行的协议,要求 Trent 分发一个好的密码证件用于接入 Bob 的服务。Trent 提供(“票证授予”)服务,该服务产生 Alice 和 Bob 共享的会话密钥,并



安全地在两个“票证”中分发会话密钥,该票证用 Alice 和 Bob 分别与 Trent 共享的长期密钥加密。这正是 Needham-Schroeder 认证协议体现的思想。

Windows 2000 是现在企业网络环境中广泛使用的一个重要操作系统,它使用 Kerberos 认证协议(基于第 5 版[168])作为该系统网络认证的基础。

Kerberos 是麻省理工学院(MIT)在 Athena 计划项目中创造的,作为其网络安全问题的解决方案。MIT 已经设计了 Kerberos 第 5 版,并将其作为免费软件(包括源码)在网上发布,可以在 MIT 网站 <http://web.mit.edu/kerberos/www/> 下载。然而,根据美国政府的出口限制,在本书写作期间,Kerberos 的可执行版只能由美国国内的美国公民或者加拿大国内的加拿大公民下载。

Kerberos 协议第 5 版比 Needham-Schroeder 认证协议(修正后的时戳版本)要稍微复杂一点。下面我们介绍本 Kerberos 协议的第 5 版。

### 12.4.1 单点登录结构

Kerberos 认证协议包括三个子协议,称为交换<sup>①</sup>。这三个交换如下:

1. 认证服务器交换(AS 交换):在“客户”C 和“认证服务器”AS 间运行。
2. 票证授予服务器交换(TGS 交换):AS 交换后,在 C 和“票证授予服务器”TGS 间运行。
3. 客户/服务器认证应用交换(AP 交换):TGS 交换后,在 C 和“应用服务器”S 间运行。

上面三个交换都是两次传输消息交换构成的协议。这些交换之间有顺序关系,该关系体现在图 12.4 的三只怪物<sup>②</sup>中。

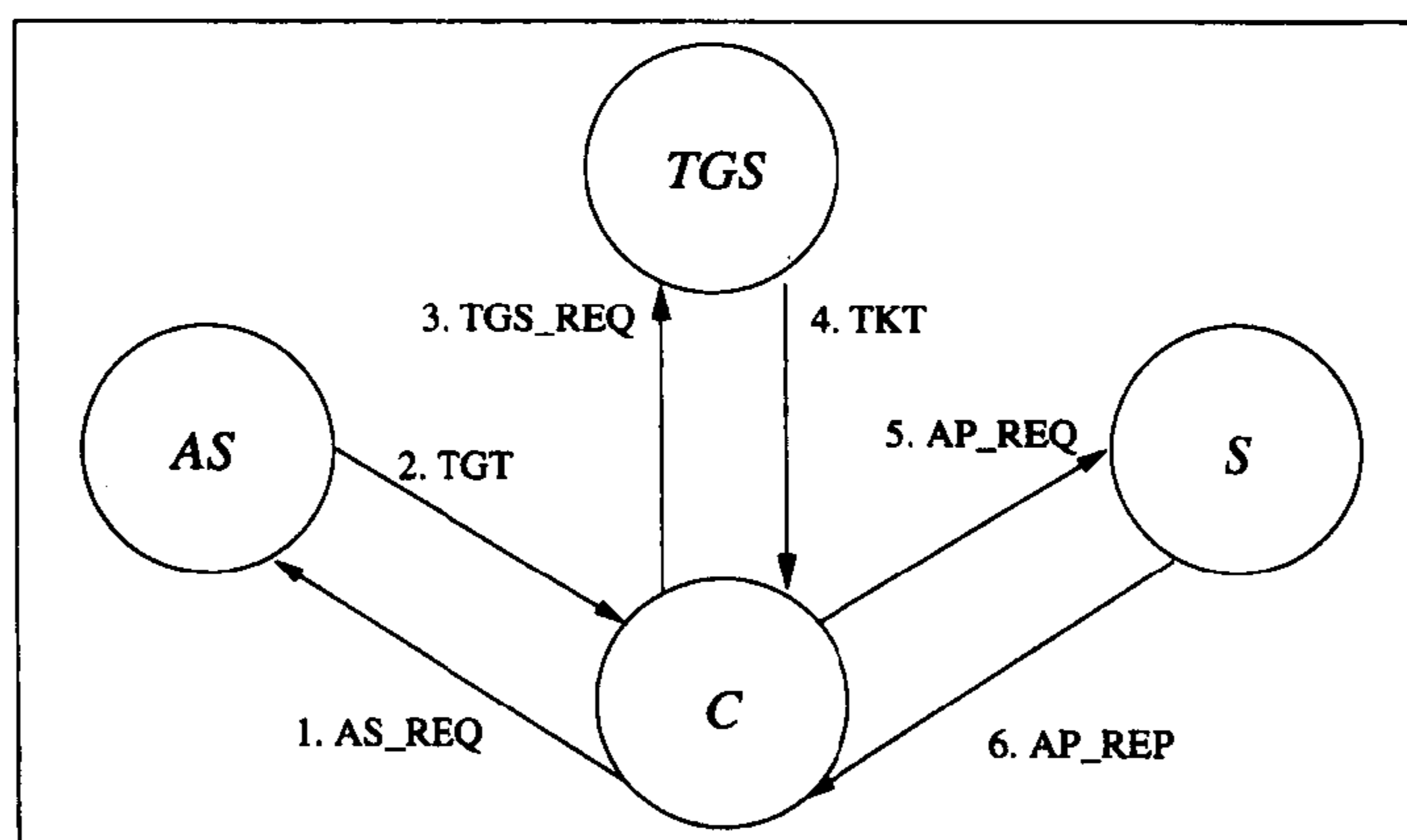


图 12.4 Kerberos 交换

Kerberos 协议中有 5 个主体参与了这三个交换,这些参与主体分别具有以下作用:

- U:用户(人类)。在协议中,用户的行为通常由客户端的进程表示;所以在协议中,U 只表现为某个消息。在使用 kerberos 系统时,每个用户都有某个口令作为其单点登录的密码证件。

① 这套协议族包含很大数目的辅助类子协议,用于完成各种具体的任务,如口令更新、票证刷新、错误处理等。然而,这里我们只描述提供认证功能的三个主要的子协议。

② Kerberos 这一名称来自于希腊神话;它是守护黄泉的三头狗。



- $C$ : 客户端(进程)。代理用户实际使用网络服务。在 AS 交换中, 用户  $U$  初始化  $C$ ,  $C$  使用  $U$  在 Kerberos 系统中的密码证件。用户的这个密码证件是在客户端进程提示  $U$  输入密码时, 用户把这个密码证件交给  $C$ 。
- $S$ : 应用服务器(进程)。向网络客户端  $C$  提供应用资源服务。在 AP 交换中, 该进程接受  $C$  发出的“应用请求”(AP\_REQ)。该进程通过“应用响应”(AP\_REP)授权  $C$  使用某项应用服务。

AP\_REQ中包含的  $C$  的密码证件称为“票证”(TKT), 而该票证中包含  $C$  和  $S$  间临时共享的应用会话密钥  $K_{C,S}$ 。

- KDC: 密钥分配中心。KDC 是以下两个认证服务器的统称:

- AS: 认证服务器。在 AS 交换中, 该服务器接收  $C$  发送的明文“认证服务请求”(AS\_REQ)。并用“票证授予票证”(TGT)作为给  $C$  的响应, 用于  $C$  接下来的 TGS 交换。初始化时, AS 与使用每个受其服务的用户分别共享某个口令。共享的口令通过单点登录方法设置, 设置方法不在 Kerberos 系统之内。AS 给  $C$  的 TGT 是 AS 交换的输出, 它由两部分组成。其一用于客户, 加密密钥从客户的单点登录口令中推导出来; 另一部分用于“票证授予服务器”(在下面的 TGS 条目中描述), 加密密钥是 AS 和“票证授予服务器”的长期共享密钥。TGT 的这两部分中都含有  $C$  同“票证授予服务器”共享的会话密钥  $K_{C,TGS}$ 。

- TGS: 票证授予服务器。在 TGS 交换中, 该服务器接收  $C$  的“票证授予请求”(TGS\_REQ)(其中包含“票证 - 授予票证”TGT)。然后响应“票证”(TKT), 这使得  $C$  可以进行和应用服务器  $S$  接下来的 AP 交换。

与 TGT 类似, TKT 也由两部分组成。其一用于客户  $C$ , 加密密钥为票证会话密钥  $K_{C,TGS}$ (在 TGT 中已经分发给  $C$  和 TGS)。另一部分用于应用服务器, 加密密钥为  $S$  和 TGS 共享的长期密钥  $K_{S,TGS}$ 。

TKT 的两个部分都含有新的应用会话密钥  $K_{C,S}$ , 由  $C$  和  $S$  共享。应用会话密钥是由  $C$  使用的密码证件, 用于  $C$  和  $S$  运行后继的 AP 交换, 用于获得  $S$  的应用服务。

#### 12.4.1.1 KDC 分为两个子服务器 AS 和 TGS 的原因

稍后我们将看到 AS 和 TGS 的作用实际上是非常相似的: 它们统称为密钥分配中心(KDC)。

把 KDC 分为作用相似的两部分, 是考虑到该系统可能会在一个相当大的网络“领域”中运行, 该领域中应用服务器属于不同的网络域, 应用服务器在不同的网络域中由不同的 TGS 统领。因此, 即使某个固定的用户只有某个固定的单点登录 AS, 他/她也可以得到多个 TGS 提供的服务, 进而得到更多的应用服务器提供的服务。

#### 12.4.2 Kerberos 交换

接下来我们分别描述 Kerberos 的三个交换。为了便于说明 Kerberos 认证协议的主要思想, 我们这里只列出 Kerberos 中必须执行的协议消息。有兴趣的读者可以参考[170]研究完全版的 Kerberos 协议, 其中描述了所有的协议消息及其细节, 包含大量的可选择消息。

### 12.4.2.1 认证服务交换

AS 交换只与  $C$  和 AS 相关:

1. AS\_REQ  $C \rightarrow AS: U, TGS, Life\_time1, N_1$

2. TGT  $AS \rightarrow C: U, T_{C,TGS}, TGT_C$

其中

$$T_{C,TGS} = \{ U, C, TGS, K_{C,TGS}, Time\_start, Time\_expire \}_{K_{AS,TGS}}$$

$$TGT_C = \{ TGS, K_{C,TGS}, Time\_start, Time\_expire, N_1 \}_{K_U}$$

消息 1 由用户  $U$  发起。客户端  $C$  使用明文消息 AS\_REQ 和认证服务器 AS 通信, 通知 AS 它将代理用户  $U$  和票证授予服务器 TGS 通信。在请求中包含生命期  $Life\_time1$  (簿记信息) 和随机数  $N_1$  (新鲜性标志符)。

在响应中, 认证服务器 AS 生成一个新的票证会话密钥  $K_{C,TGS}$ , 该密钥在  $C$  和 TGS 之间共享; 然后把生成的会话密钥加密后封装在票证-授予票证 TGT 中, 并把 TGT 作为消息 2 反向发送给  $C$ 。

TGT 中用于 TGS 的部分消息是  $T_{C,TGS}$ , 这部分消息用 AS 和 TGS 的长期共享密钥  $T_{AS,TGS}$  加密; TGT 中用于  $C$  的部分消息是  $T_C$ , 使用用户的口令  $K_U$  加密。

接收到消息 2 后,  $C$  可以对  $T_C$  解密 ( $C$  此时已经提示  $U$  输入了口令  $K_U$ ), 如果通过了所有验证 (注意验证的含义, 将在 12.4.3 节讨论), 那么  $C$  就接受票证会话密钥  $K_{C,TGS}$  和票证  $T_{C,TGS}$ 。这样,  $C$  就获得了有效的“票证授予票证”, 用于接入 TGS。

关于正确解密 Kerberos 票证的一个警告将在 12.4.3 节讨论。

### 12.4.2.2 票证授予服务交换

TGS 交换形式上和 AS 交换类似, 只是在 TGS 交换中, 用户的请求消息 TGS\_REQ 中包含称为认证者的一部分数据, 这部分数据在明文消息的后面。

3. TGS\_REQ  $C \rightarrow TGS: S, life\_time2, N_2, T_{C,TGS}, A_{C,TGS}$

4. TKT  $TGS \rightarrow C: U, T_{C,S}, TKT_C$

其中

$$T_{C,S} = \{ U, C, S, K_{C,S}, Time\_start, Time\_expire \}_{K_{S,TGS}}$$

$$TKT_C = \{ S, K_{C,S}, Time\_start, Time\_expire, N_2 \}_{K_{C,TGS}}$$

$$A_{C,TGS} = \{ C, Client\_time \}_{K_{C,TGS}}$$

这一对消息交换的功能和参与主体的行为可以参照 AS 交换类似地解释。惟一值得解释的是新增加的项  $A_{C,TGS}$ , 该数据称为认证符。其作用是向票证授予服务器 TGS 表明客户端  $C$  在  $Client\_time$  时刻使用了票证会话密钥  $K_{C,TGS}$ 。TGS 应该检查其本地主机时间并比较本地时间和  $Client\_time$ , 以确认两者的时间差在允许的范围内。

关于 Kerberos 认证者的警告在 12.4.3 节讨论。

#### 12.4.2.3 应用服务交换

最后,在 AP 交换中,客户  $C$  使用新获得的应用会话密钥  $K_{C,S}$  和票据  $T_{C,S}$  从应用服务器  $S$  那里获得应用服务。

5. AP\_REQ  $C \rightarrow S: T_{C,S}, A_{C,S}$

6. AP\_REP  $S \rightarrow C: A_{S,C}$

其中,

$$A_{C,S} = \{C, \text{Client\_time}\}_{K_{C,S}}$$

$$A_{S,C} = \{\text{Client\_time1}\}_{K_{C,S}}$$

这对消息交换的含义是明显的。

我们在前面描述两个交换时已经给出了警告,下面来具体阐述。

#### 12.4.3 警告

我们必须讨论 Kerberos 交换中的两个警告。

第一个警告是关于在解密时仔细确认 Kerberos 密文的有效性。

当主体对票证解密时,应该验证解密有效。从 Kerberos 票证的结构来看,显然该验证应该包括验证标志符的新鲜性以及验证意定主体身份的正确性。然而,还有一个不太明显的需要就是验证密文的数据完整性。在前面几章(例如 11.7.8 节)中,我们已经通过几个例子说明了对密文进行数据完整性校验的重要性。在 17.2.1 节,我们将对此做进一步的研究。

该警告适于 Kerberos 交换中所有的加密。

第二个警告是关于“认证者”。

尽管这部分数据的名称“认证符”和它的位置与用法(在票证尾部)等都表明该数据起着消息认证码的作用(MAC,见 10.3 节),可以对它前面的票证数据(例如,  $A_{C,TS}$  对它前面的  $T_{C,TS}$ )提供完整性保护。然而,这种“保护”实际上是不存在的。

我们不仅需要合适的机制来对票证提供合适的数据完整性保护(例如,通过 MAC),而且还要注意:使用加密操作来产生认证符,这本身就使用了一种错误的密码服务。为了防止对手修改认证者持有的 Client\_time,认证者的密文分组本身也需要数据完整性保护!

该警告适于 Kerberos 中的所有认证符。

### 12.5 SSL和TLS

有一个重要协议,它主要是针对万维网(Web)的安全性设计的,该协议就是安全套接层协议(SSL)[138,112]。术语“套接层”是指连接对等进程的标准通信信道,这些进程在网络设备(例如,客户/服务器计算机)中运行。套接层协议在应用层协议之下,在网络层协议之上。应用层协议包括如超文本传输协议(HTTP)、轻度目录访问协议(LDAP)或者因特网消息访问协议(IMAP)等。网络层协议包括如传输控制协议(TCP)和因特网协议(IP)。如果套接层的通信

(例如,在机密性和数据完整性方面)是安全的,那么所有应用层协议的通信至少具有同等安全性。

SSL 是一个通用的协议,用于管理因特网上消息传输的安全性。该协议最初由 Netscape 通信公司设计,并集成到该公司的 Web 浏览器(客户端软件)和 Web 服务器中。后来也被微软和其他因特网客户/服务器开发者采纳,并进而发展成为 Web 安全的事实标准,直到最近才发展成为传输层安全协议(TLS)[96]。TLS 是用于保障 Web 安全的因特网标准,由工业标准组织因特网工程任务组(IETF)制定。

TLS 是基于 SSL 设计的,并且与 SSL 相比差别不是很大。然而,因为 TLS 是 SSL 的更新版,并且是 Web 安全的国际标准,所以我们从现在开始将按照标准,在描述 Web 安全协议时只使用 TLS 这一术语。

### 12.5.1 TLS 架构概述

TLS 由两层协议组成:**TLS 记录协议**和**TLS 握手协议**。从层次关系来看,后者在前者之上。

TLS 记录协议提供通信信道的安全封装,这些信道用于高层应用协议。该协议在 TCP 和 IP 层之上,提供可靠的会话连接。该协议在发送方接收要传输的消息,把消息分为可控大小的分组,可以选择是否对数据压缩,然后附加 MAC(HMAC,见 10.3.2 节)以提供数据完整性检验,对数据加密(对称算法)保证机密性,把结果发送给对等实体。在接收方,接收到密文分组后,解密这些分组,验证 MAC,可以选择是否需要解压缩,重组分组,最后把结果提交给高层应用进程。

每次会话中对称加密和 HMAC 所需的密钥是惟一生成的,密钥的获取基于 TLS 握手协议的秘密协商。

TLS 握手协议允许服务器和客户相互认证,协商密码算法,确认密钥并在确认的密钥基础上建立安全会话连接,该安全连接用于 TLS 记录协议处理高层应用协议的安全通信。

从 TLS 架构的描述中可以看出,虽然 TLS 记录协议是获得安全通信必不可少的一部分协议,但是该协议并不是认证协议。因此,我们将只介绍 TLS 握手协议。

### 12.5.2 TLS 握手协议

可以认为 TLS 握手协议是有状态的进程,该进程运行在客户和服务器计算机上。一个有状态的连接称为一次“会话”,其中通信对等方分别执行以下操作:

- 交换 hello 消息用于协商算法,交换随机数,验证是否是会话重复连接。
- 交换必需的密码参数,使得客户端和服务端能够协商某个密钥(称为“主密钥”)。
- 交换证书和密码信息,使得客户端和服务端能够把自己向另一方认证。
- 通过交换随机数,由主密钥生成会话密钥。
- 验证它们的对等方已经计算了相同的安全参数,确认握手已经完成,并且未遭受到攻击者的篡改。
- 把建立起来的安全信道提交给 TLS 记录层,用于处理高层应用通信。

这些步骤通过我们下面描述的四次消息交换来实现。为了更好地说明该协议,我们将只描述 TLS 握手协议的一个简化版本,该版本忽略了一些可选项。协议描述中,C 代表客户端(例如客户端的 Web 浏览器),S 代表 Web 服务器。如果消息后面附有 \* 号,该消息就是可选的。

1.  $C \rightarrow S$ : ClientHello;
2.  $S \rightarrow C$ : ServerHello,  
ServerCertificate \*,  
ServerKeyExchange \*,  
CertificateRequest \*,  
ServerHelloDone;
3.  $C \rightarrow S$ : ClientCertificate \*,  
ClientKeyExchange,  
CertificateVerify \*,  
ClientFinished;
4.  $S \rightarrow C$ : ServerFinished.

在 ClientKeyExchange 消息和所有可选择消息忽略时,该协议也能够运行。这就是当客户希望重新开始某个存在的会话时的情形。

下面我们概述性解释 TLS 握手协议中交换的消息。

#### 12.5.2.1 Hello 消息交换

客户端发送 ClientHello 消息发起会话连接,接收该消息的服务器必须用 ServerHello 消息响应,否则连接失败。这两条消息包含以下域:“protocol\_version”、“random”、“session\_id”、“cipher\_suite”和“compression\_methods”。

“protocol\_version”用于后向兼容:服务器或客户端使用该域通知对方本方使用的协议版本。“random”包含随机数(一次性随机数作为新鲜性标识符)。该数据由通信双方各自产生,并进行交换。该域也包含每个通信方的本地时间。

“session\_id”惟一标志当前会话连接。当客户希望开始新的会话连接时,ClientHello.session\_id 域应该为空。这时,服务器会生成一个新的会话 id,在本地内存中缓存该会话 id,并在 ServerHello.session\_id 域中使用该值。如果 ClientHello.session\_id 域非空(当客户端要重新开始某次存在的会话时),服务器应该在其本地缓存中寻找该会话 id,重新开始客户端要求的会话连接。

需要注意的是“cipher\_suite”域。ClientHello.cipher\_suites 是客户端计算机支持的一些密码选项,这些选项按照客户端最想用的顺序排列。客户端可以提出系列的公钥和对称密码算法、数字签字机制、MAC 机制和杂凑函数。对每一种必需的密码操作服务器挑选某种配置,然后在 ServerHello.cipher\_suites 中通知客户端。

#### 12.5.2.2 服务器证书和密钥交换数据

如果客户端要认证服务器,在 hello 消息交换之后,服务器应该选择发送其证书。ServerCertificate 消息如果非空,包含的就是系列的 X.509.v3 证书(见 13.2 节)。X.509 证书中包含了足够的关于姓名和证书所有者公钥以及签发证书机构的信息(见例 13.1)。服务器发送系列证书是为了使客户可以选择客户端计算机支持的某个证书,并使用规定的公钥算法。

ServerCertificate 域之后是 ServerKeyExchange 域。该域包含了服务器的公钥素材,这些素材与 ServerKeyExchange 域中的服务器证书匹配。用于 Diffie-Hellman 密钥协商的素材在需要时也

包含在该域中,以三元组 $(p, q, g')$ 的形式出现,其中  $p$  是大素数模,  $g$  是大的模  $p$  群的生成元,  $y$  是服务器本地缓存中的整数(该数与“session\_id”相关)。

服务器如果不提供匿名服务,并且与在 ClientHello.cipher\_suites 中选择的公钥算法不冲突,那么该服务器就可以使用 CertificateRequest 域进一步要求客户提供其证书。

最后,服务器发送 ServerHelloDone 消息,指示握手的 hello-message 阶段结束。接下来服务器等待客户响应。

### 12.5.2.3 客户响应

如果服务器发送了 CertificateRequest 消息,客户端要么发送 ClientCertificate 消息,要么发送 NoCertificate 的警告消息。

接下来发送 ClientKeyExchange 消息的具体内容取决于 ClientHello 和 ServerHello 消息中协商的公钥算法。

如果客户端的 KeyExchangeAlgorithm 是 RSA,客户端就生成“主密钥”(48 字节值),然后使用服务器证书的 RSA 公钥加密(从 ServerCertificate 中获得)。

如果客户端发送了证书并且客户端有签名的能力,那么客户端就会向服务器发出带有数字签名的 CertificateVerify 消息,用于服务器明确验证客户端的证书。

### 12.5.2.4 最后的消息交换

客户端发送 ClientFinished 消息,该消息中包含带有密钥的 HMAC(基于“主密钥”加密),以便允许服务器确认客户端执行了正确的握手程序。

作为响应,服务器发送其 ServerFinished 消息,其中也包含带有密钥的 HMAC,以便允许客户端确认服务器执行了正确的握手程序。

上述交换完成之后,握手完成,客户端和服务器可以开始交换应用层数据。

## 12.5.3 TLS 握手协议的典型运行

下面给出握手协议的一个典型运行,以完成我们对 TLS 协议的描述。该运行的例子在协议 12.2 中给出。

---

### 协议 12.2 TLS 握手协议的典型运行

1.  $C \rightarrow S$ : ClientHello. protocol\_version = “TLS Version 1.0”,  
ClientHello.random =  $T_C, N_C$ ,  
ClientHello.session\_id = “NULL”,  
ClientHello.crypto\_suite = “RSA: encryption, SHA-1: HMAC”,  
ClientHello.compression\_method = “NULL”;
2.  $S \rightarrow C$ : ServerHello. protocol\_version = “TLS Version 1.0”,  
ServerHello.random =  $T_S, N_S$ ,  
ServerHello.session\_id = “xyz123”,  
ServerHello.crypto\_suite = “RSA: encryption, SHA-1: HMAC”,  
ServerHello.compression\_method = “NULL”;



```

ServerCertificate = point_to(server's certificate),
ServerHelloDone;
3.  $C \rightarrow S$ : ClientKeyExchange = point_to(RSA_Encryption(master_secret));
   ClientFinished = SHA-1(master_secret || C || ,  $N_C, N_S, \dots$ );
4.  $S \rightarrow C$ : ServerFinished = SHA-1(master_secret || S || ,  $N_S, N_C, \dots$ )。

```

在 TLS 握手协议执行过程中,客户端如果选择匿名,就不必向服务器认证。客户端选择使用 RSA 加密,SHA-1 计算 HMAC。这样,该协议完成的是服务器向客户的单方认证。协议执行后的输出是服务器向客户端的一个单方认证信道。

上述运行是使用 TLS 协议的一个典型例子,应用在基于 Web 的电子商务中,例如从在线书店购书。协议输出的单方认证信道可以向客户保证下面这一点,就是只有认证的服务器才能够接收它的购书指令,该指令中可能包含机密信息,例如用户的银行卡信息、书名和交货地址。

#### 12.5.4 对 TLS 协议的边信道攻击

在边信道攻击中,Malice 寻求的是一些潜信道信息,这些信息是合法主体无意间泄漏的。定时分析攻击是边信道攻击的一种典型情况。这种攻击中,Malice 观察并分析某个主体对其询问的响应时间,以便获得某种秘密信息。首先发表边信道和时间分析攻击的是 Kocher [169],该攻击对于执行模指数运算的系统非常有效(例如,RSA 签名或解密,ElGamal 类算法的临时密钥求幂,Diffie-Hellman 密钥交换签名机制)。攻击的目标是恢复秘密指数。模指数运算使用平方-乘方法对指数进行逐比特运算(见算法 4.3)。该运算对于指数中的比特 1,执行的操作是平方和乘,而对于指数中的比特 0,则只执行平方操作。该攻击的核心就是检测这两种操作的时间差。成功的检测意味着可以逐比特地得到秘密的指数。

最近,Canvel 等[69]发现了对于协议的边信道攻击(使用定时分析)技术,其攻击对象是客户和服务端间连接由 TLS/SSL 保护的链路。该攻击的典型目标是用户访问电子邮件(IMAP)服务器所用的口令。该口令是在 TLS 链路保护的客户端和服务端间的通信信道上传输的,传输方向为从客户计算机发送到邮件服务器。该通信信道用强的会话密钥加密,该会话密钥是从 TLS 协议得到的(例如,协议 12.2 所说明的协议)。会话加密使用的算法是强的分组加密算法(例如三重 DES),加密算法的运行模式是 CBC 模式(见 7.8.2 节)。

时间分析攻击利用了 Vaudenay 对于标准 CBC 填充机制的“炸弹式预言机攻击”[296]。我们在 7.8.2.1 节已经对该攻击做了介绍。现在我们简单回忆该攻击的流程。设  $C$  是 CBC 密文分组,其加密对象是口令,并且 Malice 记录了这个分组。在 Vaudenay 对于标准 CBC 填充机制的攻击中,Malice 发送以下内容给解密预言机

$$r, C$$

其中  $r$  是随机的单个或多个数据分组。Malice 接下来等待解密预言机的应答,要么是“正确的填充”,要么是“不正确的填充”。“正确的填充”应答就向攻击者泄露了  $C$  加密的最后一个字节(如果  $C$  的加密对象是口令,那么泄露的这个字节就是口令的最后一个字符)。有了上述技术基础,我们就可以描述对于 TLS 连接的时间分析攻击了。

Malice 假装是  $C$  加密的口令的所有者,并且要访问邮件服务器,向邮件服务器发送  $r, C$ 。服务器接收到  $r, C$  后,执行 CBC 解密验证填充的正确性。如果填充是正确的(概率约为  $2^{-8}$ , 见 7.8.2.1 节),服务器就重构 MAC(消息验证码,见 10.3.3 节)验证数据完整性。如果检测到填充错误,那么就没有必要执行数据完整性校验(不必重构 MAC)。验证结果或者是填充错误或者是 MAC 错误,都会通过 TLS 强的会话密钥加密后送给客户端。

表面看起来因为 Malice 不知道强的会话密钥,不能得到预言机服务,也就是说,电子邮件服务器发送的错误消息是加密的,因此不是解密预言机。

然而,因为  $r$  是随机的,如果 CBC 填充是正确的,那么数据完整性校验会以“压倒性”概率失败。因此,受攻击邮件服务器实际上只会给出以下两种响应:

- i) 响应 {“无效填充”} $_K$ , 概率约为  $1 - 2^{-8}$ ;
- ii) 响应 {“无效 MAC”} $_K$ , 概率约为  $2^{-8}$ 。

情形(ii)意味着“正确的填充”,此时 Malice 得到了  $C$  加密的明文的最后一个字节。

现在使用定时分析攻击! 对于一个足够大的  $r$ (占用几个分组),对于情形 ii),邮件服务器不得不重新计算一个很长的 CBC MAC;而对于情形(i),邮件服务器则不涉及这样的计算。如果服务器使用标准的 CBC 模式,Canvel 等[69]检测到两种情况下服务器的响应时间的确存在一个固定的差别,这种差别大约为几个毫秒。这样,在时间分析下,服务器确实是一个解密预言机。注意到错误处理程序在应用中通常是必需的,这就意味着这种解密预言机不会爆炸,它是一个可靠的预言机!

Malice 通过机智地改变  $r$ (不改变  $C$ ),就可以用逐字节追溯的方法获得整个口令。改变  $r$  的方法留给读者作为习题(在习题 12.12 中有提示)。如果  $C$  加密的口令是 8 字节,提取整个口令只需  $8 \times 2^8 = 2048$  次试验,每一次试验都通过假装成一次电子邮件访问登录会话完成。

这是一次非常规攻击,在客户和服务器的同一个本地网(LAN)的情况下,该攻击会更有效(可能也仅限于此种情形),因为这时延迟时间能够被精确地检测到。该攻击显示预言机服务通常是可以得到的,有时能够通过边信道获得。从该攻击中,我们也会看到应当谨慎处理密码协议中的错误消息。

对这种攻击的一种可能的修正方法是服务器在响应错误消息前应该进行一段随机时间的“睡眠”。

## 12.6 本章小结

本章,我们介绍了四个认证协议(系统和标准)的实际应用。包括 IKE,IPSec 的 IETF 认证标准;SSH,安全壳交互会话远程接入的事实认证标准;Kerberos,基于视窗操作系统用于访问企业计算机和信息资源的工业标准;TLS(SSL),Web 安全的事实标准。

尽管我们在描述每个协议族(系统)时,都做了很大程度的简化,但我们的描述仍然显示出很大的工程复杂性。这种复杂性是由实际系统的需求引起的,例如算法和参数的协商,多种不同系统的兼容,后向兼容,易于使用等。对于 IPSec 和 IKE 来说,要求支持机密性的一般性质也增加了系统的复杂性。通过本章的研究可以看到,认证协议在实际应用时,我们不仅要面对系列的安全问题,还要面对大量的系统工程问题。而工程问题一旦处理失误,会造成严重的安全后果。

我们也看到认证协议的极易出错的本质,在实际应用中会不可避免地显现出来。所以,到目前为止我们仍然未能完成本书认证协议这一主题。在第 17 章介绍形式化分析技术时,我们会再次回到这一重要主题。

## 习题

- 12.1 在 IP 通信中,如果缺少 IPSec 的保护,Malice 可以怎样操纵在因特网上传输的消息(例如伪装消息的生成者,按新的路线发送消息等)?
- 12.2 在有 IPSec 保护的 IP 数据报中,“认证头”(AH)起什么作用?
- 12.3 IPSec 和 IKE 的关系是什么?
- 12.4 使用哪两种方法可以使 IP 数据包得到密码保护?
- 12.5 在习题 11.15 中,我们已经得到一种在修正 STS 协议的小缺陷的同时不损害其匿名(可否认性)特征的方法。请提供一个类似的修补方法,用于 IKE 基于签名的阶段 1 主模式,修正其中小缺陷的同时不损害其“看似合理的拒绝”这一属性。
- 12.6 给出 12.2.3.4 节基于签名的 IKE 阶段 1 进取模式的“完善拒绝服务攻击”的攻击流程。提示:该攻击与攻击 11.3 类似。
- 12.7 加密的密钥交换(EKE)协议(协议 11.5)和 SSH 协议都使用非对称加密算法加密口令,然而它们存在本质区别,请指出该区别。
- 12.8 在实际中,SSH 协议中的服务器如何向客户端的用户认证?
- 12.9 为什么在 Kerberos 协议的一般设定中,每一个客户端都要面临三种不同类型的服务器?
- 12.10 为什么 Kerberos 协议适合在企业内部环境中应用?该协议是否适用于在企业之间(开放系统)的环境中应用?
- 12.11 TLS(SSL)协议在基于 Web 的电子商务应用中广泛使用。这些协议是否自然地适于这样的应用环境?如果不是,为什么?  
提示:这些协议不支持带有不可否认服务的支付授权。
- 12.12 在 12.5.4 节,我们介绍了一种定时分析攻击技术,用于获取明文消息的最后一个字节,该明文消息加密后的密文在 CBC 密文分组中,加密时使用了标准的 CBC 明文填充方式。那么,如何获得明文消息的其他字节呢?  
提示:回忆 7.8.2.1 节标准的 CBC 明文填充方式;在成功提取最后一个字节后,为了提取最靠近该字节的前一个字节,应该考虑到“正确填充”的以下事件:最后两个字节(“两个填充字节”)是‘02’ || ‘02’;考虑修改  $r$  的最后一个字节,使该事件发生的概率尽量大。

## 第 13 章 公钥密码的认证框架

### 13.1 前言

就通常的公钥密码而言,密钥的生成过程始终包含如下步骤:

$$\text{公钥} = F(\text{私钥}) \quad (13.1.1)$$

其中  $F$  是一个从私钥空间映射到公钥空间的有效单向函数。由于函数  $F$  的单向性(一个好的混合变换),由私钥计算所得的公钥总包含一段看似随机的成分。

由于每一个公钥都包含着一段看似随机的成分,显然有必要让主体(用户)的公钥以一种可验证的和可信的方式与主体(用户)的身份信息相关。很显然,为了传递一条用公钥加密的秘密消息,发送者必须确信这个看似随机的公钥确实属于所声称的签名人。

通常,在实际应用中为了应用公钥密码系统,我们需要一个能够简单验证公钥与主体身份相关的验证机制。这样的机制可以在认证框架中实现:公钥的拥有者可以向系统认证。

#### 13.1.1 本章概述

在本章的以下部分,我们将介绍建立公钥密码系统认证框架的两种不同方法:一种称为公钥证书基础设施(PKI),见 13.2 节;另一种称为基于身份的公钥密码学,见 13.3 节。

### 13.2 基于目录的认证框架

对于一对频繁通信的主体而言,他们需要能够容易、安全地识别对方的公钥:起初可以以一种安全的物理方式交换他们的公钥,如面对面地交换并以一种安全的方式保存密钥。然而,这种“简单”密钥管理方法的扩容能力并不理想。在一般开放的通信系统中,通信双方可能事先并不认识,并且在大多数情况下,双方只通信一次。这种“简单”的密钥管理方法要求每个主体都要管理不现实的巨大数量的公钥。另外,这种方法也不能充分利用公钥密码学的优点。

在 2.4 节我们已经看到由一个可信主体提供的密钥管理的在线服务。这个服务由几个子服务组成,如密钥登记、认证和姓名目录表。为了使用密钥管理服务,每个主体通过与可信的主体(认证服务器)分享长期密钥来建立一个长期——对应的关系。当在两个主体(端用户)之间需要进行安全通信时,他们与认证服务器一起运行一个认证协议,在他们之间建立起一条安全的通信信道。因此,每个端用户主体仅仅需要管理一个与认证服务器共享的密钥即可。第 2 章介绍的密钥管理和认证协议是基于密钥的密码系统(尽管在 2.6.6 节我们讨论了 Needham-Schroeder 公钥认证协议,在该协议中的认证服务也是使用在线可信第三方,其本质是一种密钥形式)。

密钥管理服务可以很自然地扩展到公钥管理,这里的密钥管理称为公钥证书服务,并且可信的服务器称为证书机构(CA)。CA 是一个特殊的主体,在其服务域内的所有主体都信任他,并通过间接的方式(我们将讨论更多的信任问题)被更大域内的主体知道和信任。对于 CA 域内每个端用户而言,CA 发放一个公钥证书来证实用户的公钥数据。公钥证书是一个包含多项

数据的结构化数据记录,它包括持有者惟一可识别的身份、公钥参数以及 CA 的签名。CA 对证书的签名提供了持有者的身份与她/他的公钥的密码绑定。如果一个主体信任 CA,那么,当他验证了另一个主体的证书有效后,她/他就应该信任这种绑定的有效性,这是因为只有当 CA 正确地识别了持有者以后,才发放证书。这样,验证主体通过公钥就建立起一条由证书到她/他(实上是到系统)的安全密钥信道。Kohnfelder 首先使用“公钥证书”这个概念[171]。

如在图 7.1 和图 10.1 所描述的,基于证书服务的公钥信道通常称为**基于目录的信道**。因而,证书服务也称为目录服务。

注意,与基于认证协议的密钥认证服务器所要求的“信任”相比(见 2.4 节),CA 所要求的“信任”就弱得多了。在这里,所提供的安全服务是消息的认证,这个认证不需要处理任何秘密(由于验证只需要 CA 的公钥就可以验证 CA 的签名)。于是,这个服务可以是离线的,也就是说,CA 不需要和端用户一起运行协议。离线服务的一个重要特点是,它可以通过增大规模来处理大规模系统。显然,用 CA 的公钥来验证由它所发行的证书,反过来,它的公钥也可以包含在另一个 CA 所签发的证书中,依次类推。

证书中的数据项应该包括身份消息和发行 CA 的公钥信息。这些数据项也应该包括一些附加的信息,如描述对公钥证书 CA 签名的验证算法、有效期和使用条件等。一个非正式公钥证书的定义如例 13.1。

#### 例 13.1 公钥证书

```
Certificate::=
{
    issuer name;
    issuer information;
    subject name;
    subject information
    validity period;
}

Issuer information::=
{
    issuer public key;
    signature algorithm identifier;
    hash function identifier;
}

Subject information::=
{
    subject public key;
    public key algorithm identifier
}

Validity period::=
{
    start date;
    finish date;
}
```



### 13.2.1 证书发行

发行证书时,CA 应该核实请求证书主体身份的有效性。当然这类核实应该包括一些物理的(非密码的)识别方法,和我们通常在一些商务交往中所做的一样(如在银行开账户)。主体也应该证明她/他知道要签发的公钥对应的私钥。这种证明可以以用户生成对一条询问消息签名的形式来证明,其签名可以用公钥来验证,或者以用户和 CA 之间进行零知识证明的形式。某些应用要求公钥的秘密部分具有特定的结构。在这些应用中,零知识协议可设计成能够证明所需要的结构。在第 18 章我们将看到几个用来证明这类秘密结构的零知识证明协议。

### 13.2.2 证书吊销

有时候需要吊销某个证书,如当用户私钥被泄露或用户的信息发生了变化时,都需要吊销证书。

对于基于目录表的证书框架的情况,根 CA 应该维护一个吊销证书的最新列表,这个最新列表应该是在线可用的。另一种方法是,根 CA 可以在整个系统中发布一个“ $\Delta$  吊销列表”,该列表仅仅包含最近被吊销的证书。当系统范围内的用户收到  $\Delta$  吊销列表后,就能够更新他们的本地证书吊销列表副本。

吊销的证书应该由吊销 CA 加盖时戳。即使对签名的验证时间是在证书吊销之后,在证书吊销时间以前主体所生成的签名应仍然有效。

### 13.2.3 公钥认证框架实例

现在让我们看几个基于目录表的公钥认证框架实例。

#### 13.2.3.1 X.509 公钥证书框架

标准的公钥证书框架称为 X.509 公钥证书基础设施[154],规模呈树状层次结构增大,称其为**目录信息树(DIT)**。在这种树状层次结构中,每个节点代表一个主体,并且他的公钥证书由它相邻的父节点发行。叶节点表示端用户主体,非叶节点代表不同级别或不同域的 CA;例如,国家级 CA 有工业、教育和政府机构域;其中每一个域又分为许多子 CA,例如教育域又分为不同的大学子域。根节点代表**根 CA**,它是整个系统中众所周知的主体。根 CA 应该保证他自己的公钥的可靠性。因为每个 CA 都有服务于一个很大的(CA 或终端用户的)域的潜力,所以 DIT 的深度不需要很大。两个端用户主体通过在 DIT 中向上找一个离他们最近的父节点来建立一条安全的通信信道。

#### 13.2.3.2 PGP“信任网”

另一个拥有大量业余用户的公钥证书框架称为 PGP“信任网”或“密钥环”(PGP 是 Pretty Good Privacy 的缩写,是 Zimmermann[314]提出的一种安全电子邮件软件)。这种认证模型的规模按非层次结构方式增大。在 PGP“信任网”中,对于这个系统中的其他主体而言,任何个人都可通过对他们的“密钥证书”,即一个 $\langle \text{name}, \text{key} \rangle$ 对进行签名而成为“CA”。显然,这种签名关系形成一个网状结构。在这个网中的任一“CA”都不是完全可信赖的,或根本不能信赖。PGP 所依据的理论是,拥有足够多的这种签名,就可以信任这种 $\langle \text{name}, \text{key} \rangle$ 结合,这是因为并非所有的签名者都是坏人。因此,当 Alice 需要证实 Bob 密钥的真实性时,她应该请求审阅 Bob 的



多个密钥证书。如果 Alice 在“某种程度”上“了解”这些证书的某些发行“CA”，那么她就得到了有关 Bob 公钥的一定程度的真实性。Alice 可以要求 Bob 提供更多的“证书”，直到她对信任程度满意为止。

### 13.2.3.3 简单公钥基础设施

X.509 公钥证书框架可以看做是一个全球在线的电话簿。每个用户在其中都有一个记录，因此在每个用户证书中的 Subject name 必须是一个全局可区分的名字(见例 13.1)。这样的认证框架似乎足以满足公钥密码的早期应用：机密性的安全通信(抗窃听)，一条机密消息的接收者应该用他的密钥惟一区分。

自 1990 年以来，公钥在电子商务、远程访问和许多活动中得到越来越广泛的应用(参见前言中列举的应用)。Ellison 等人考虑到一些新出现的应用，认为一个全球的具有绑定密钥的可区分名字越来越变得不适用了[104]。当给定一个公钥证书后，应用所要做的就是回答是否允许远程密钥持有者进行某种访问或某种授权操作。应用必须作出一个决定。作出决定所需要的数据几乎从来都不是密钥持有者名字的拼写，而是这些应用需要知道密钥的持有者是否被授权做某种访问。这应该是公钥证书的基本工作。

Ellison 等人也认为最初的 X.500 计划永远不大可能成熟起来。目录表(如雇员列表、用户列表、通信列表等)的汇集对于拥有者来说是有价值的，或者甚至是机密的，故不太可能以 X.500 目录的子树的形式公诸于众。举一个极端的例子，他们想像 CIA 会将它的代理目录表加到世界范围 X.500 的库中，这怎么可能呢？一个可区分名字的 X.500 想法(当查阅一条目时，每个人都能够使用全局惟一的名字)也不大可能出现。这种想法要求全局命名规则，并且已有的太多条目并非在单一的命名规则下定名。因此，这种遗产妨碍了这种想法。

Ellison 等人提出了一个基于目录的公钥证书框架，称为**简单公钥基础设施 SPKI**[104]。它也是一个树状结构的框架，与 X.509 公钥证书框架相似。然而，它的命名规则是包含一个人的名字和他的公钥的一个杂凑值。例如：

```
(name (hash sha1|TLCgPLF1GTzgUbcaYLw8kGTENUk = 1) jim therese)
```

就是一个名字为“Jim Thereses”的人的专用 SPKI 名字。其中，采用公钥的 SHA-1 杂凑函数值使得 SPKI 名字具有全局惟一可识别性，尽管有许多人的名字都叫“Jim Thereses”。

这种命名方法由 Rivest 和 Lampson 在**简单分布式安全基础设施(SDSI)**[247]中建议。SDSI 起着局部命名规则的重要作用。这些作用仍然针对构造非集中式认证和授权框架。因此，SPKI 名字也称为 SDSI 名字。

SPKI 也考虑了持有授权和委托信息的“授权”和“委托”实体。一条授权消息可能是绑定在公钥上的一个授权描述。因此，证书能直接显示请求者是否被授权进行某一行动。委托消息描述请求者对另一个人委托授权的权利。我们可以说 SPKI 将 X.509 认证框架扩展到具有授权和委托功能的 X.509 认证框架。SPKI 认证方案的核心是应用类 LISP<sup>①</sup> 语言和 Rivest[246]提出的 S 表达式。一个 S 表达式的例子如下：

```
(object document(attributes(name *.doc)(loc Belgium))
  (op read) (principals (users OrgEU)))
```

① LISP:一种程序设计语言。

它表示在 OrgEU 中所有用户被授权阅读的对象是文档类型,且该文档的扩展名是 doc 并且位于 Belgium(比利时)。

PolicyMaker[41]是另一个在认证框架中涉及授权和政策问题的提案。PolicyMaker 起着描述证书持有者的角色和基于角色策略的重要作用。

#### 13.2.4 与 X.509 公钥证书基础设施相关的协议

在 X.509 公钥证书基础设施中,有几个与实际问题相关的协议:

- 证书管理协议(CMP)[7,210]。这个协议支持公钥基础设施(PKI)各部分之间在线交互。例如,用于证书机构(CA)与密钥对相关的用户系统之间,或者两个相互分发证书的 CA 之间的管理协议。当一个实体要求得到密钥证书或更新密钥时,他需要证明拥有私钥,此时需要用到这些交互作用。
- 在线证书状态协议(OCSP)[209]。该协议能够确定一个已识别证书的(吊销)状态。OCSP 可以用来满足一些支持比吊销列表更及时的吊销信息操作要求,也可用来获得附加的状态消息。OCSP 客户向 OCSP 回应者发布一个状态请求,在收到回应者的应答之前暂缓证书的吸收。
- 因特网 X.509 公钥基础设施时戳协议(TSP)[6]。该协议由一个发向时戳服务器的请求和时戳服务器(TSA)返回的应答组成。就处理要产生应答的请求而言,它给出了几个与 TSA 操作安全相关的要求。不可否认服务要求在指定时间以前建立数据的存在性。这个协议可用做支持此类服务的基本构件。
- 因特网 X.509 公钥基础设施运行协议(OP):FTP 和 HTTP [142]协议。这是一个使用文件传输协议(FTP)和超文本传输协议(HTTP)从 PKI 资料库来获得证书和证书吊销列表(CRL)协议规范的说明。

这些协议在 IETF 标准化主体“公钥基础设施 X.509 工作组”下发展成标准(PKIX 工作组)。我们没有描述这些协议的详细内容。有兴趣的读者可以访问 PKIX 工作组的网址:

<http://www.ietf.org/html.charters/pkix-charter.html>

从中可以下载描述这些协议(上面曾引述)的文档。

### 13.3 基于非目录的公钥认证框架

在式(13.1.1)中的一般公钥密码学意义下的密钥生成过程导致了所有公钥的随机化。因此,在认证过程中,把一个主体的公钥与他的身份消息结合起来是十分必要的。我们可以看到这样的结合可以通过一个公钥认证框架来实现:树状层次公钥证书基础设施(如 X.509 公钥证书框架,见 13.2.3 节)。然而,为了建立和维护这种树状层次结构,PKI 会导致系统异常复杂且成本过高。人们一直希望标准的公钥认证框架能够简化。

有理由认为,如果公钥看起来不随机,那么这个系统的复杂度和建立与维护该公钥认证框架的代价就会减小。可以想像,如果一个主体的公钥本身显然与该主体的身份信息如名字、电子邮件和邮政地址等附属信息相联系,那么,在本质上就不需要认证该主体的公钥。的确,我们的邮政系统就是按这种方式工作的。

Shamir[262]以一种异乎寻常的悟性开创了一个公钥密码体制。该体制能大大减小密钥认证系统的复杂度:本质上与邮件系统相似。在这个不寻常的公钥密码体制中,密钥的生成过程有如下步骤:

$$\text{私钥} = F(\text{主密钥}, \text{公钥}). \quad (13.3.1)$$

该密钥生成步骤所采取的方向与通常公钥密码体制下密钥生成的步骤相反,见(13.1.1)。当然,为了使所计算的私钥保密,这个计算过程不能公开,只限于特许的主体(如可信机构 TA)知道。为了能完成式(13.3.1)的计算,TA 拥有专有的主密钥。既然公钥作为密钥生成过程的输入,那么任意比特串就都可以作为公钥! 因为用身份消息作为公钥大大减少了公钥认证的复杂性。Shamir 在他的新公钥密码体制中建议用户的身份可以作为公钥,因此,他将他的新方案命名为**基于身份的公钥密码学**。

很显然,式(13.3.1)中的密钥生成过程是 TA 提供给系统用户的一种服务。这种服务本质上是一种认证服务:TA 根据主体的身份 ID 作为公钥为其生成私钥,用她/他的身份作为公钥使得系统中的任何用户都能识别并使用它。在为主体生成私钥以前,TA 应该对主体的身份消息进行一次全面的检验。该检验应该包括物理的(非密码的)验证方法。并且,主体提供的身份消息必须使 TA 相信该身份消息能够惟一地识别该主体。在 CA 发放公钥证书给主体以前,一个类似的识别检验是必要的(见 13.2.1 节)。

既然用户的私钥是由 TA 生成的,那么他们必须绝对、完全、无条件地信任 TA,即他们必须容忍以下情况:TA 能够阅读他们的秘密通信或伪造他们的签名。因此,基于身份的密码应该仅适合于对于用户来说无条件信任是可以接受的应用。在一个机构环境下,如果雇主拥有雇员接收和雇员发出的信息的完全所有权,那么雇主就可以充当 TA 的角色。然而,TA 可能代表多个共同为一个用户计算式(13.3.1)的实体。那么,秘密入侵必须由这些实体共同进行。这种汇集的信任的基础更容易被接受。我们将在 13.3.7.1 节中看到这样的技术。

由于主体的惟一可以识别的身份作为他的公钥,在基于身份的公钥密码系统的应用中,不需要用户建立一条密钥信道,即图 7.1 和图 10.1 中的密钥信道不再需要了。此外,图 7.1 中的  $k_e$  和图 10.1 的  $k_v$  可以用一条不证自明的信息来代替,如一个全局可区分的身份。

### 13.3.1 Shamir 的基于 ID 的签名方案

在 Shamir 的基于身份的签名方案中有 4 个算法:

- **建立:** 这个算法由 TA 运行(从现在开始我们称 TA 为 Trent)来生成系统参数和主密钥。
- **用户密钥的生成:** 这个算法也由 Trent 执行,输入主密钥和一条任意的比特串  $id \in \{0,1\}^*$ , 输出与  $id$  对应的私钥;这是式(13.3.1)的一个实例。
- **签名:** 一个签名算法;输入一条消息和签名者的私钥,输出一个签名。
- **验证:** 一个签名的验证算法;输入一个消息-签名对和  $id$ ,输出 True 或 False。

算法 13.1 具体描述了 Shamir 的基于身份的签名方案。

---

#### 算法 13.1 Shamir 的基于身份的签名方案

##### 系统参数的建立

Trent 建立:

1.  $N$ : 两个大素数的乘积;
2.  $e$ : 一个整数且满足  $\gcd(e, \phi(N)) = 1$ ;  
(\*  $(N, e)$  是系统范围内用户采用的公开参数 \*)
3.  $d$ : 一个整数且满足  $ed \equiv 1 \pmod{\phi(N)}$ ;  
(\*  $d$  是 Trent 的主密钥 \*)
4.  $h: \{0, 1\}^* \mapsto \mathbb{Z}_{\phi(N)}$ ;  
(\*  $h$  是一个强单向杂凑函数 \*)

Trent 秘密保存系统的私钥  $d$  (主密钥), 并公开系统参数  $(N, e, h)$ 。

### 用户的密钥生成

假设 ID 表示用户 Alice 惟一可以识别的身份。在进行 Alice 身份的物理验证和确认 ID 具有惟一性之后, Trent 生成密钥如下:

$$g \leftarrow \text{ID}^d \pmod{N}$$

### 签名的生成

为了对一条消息  $M \in \{0, 1\}^*$  签名, Alice 随机选择一个数  $r \in_U \mathbb{Z}_N^*$ , 并计算

$$\begin{aligned} t &\leftarrow r^e \pmod{N} \\ s &\leftarrow g \cdot r^h(t \parallel M) \pmod{N} \end{aligned}$$

所得到的签名为  $(s, t)$  对。

### 签名的验证

已知消息  $M$  和签名  $(s, t)$ , Bob 用 Alice 的身份 ID 按以下过程验证签名的正确性:

如果  $s^e \equiv \text{ID} \cdot t^{h(t \parallel M)} \pmod{N}$ , 那么,  $\text{Verify}(\text{ID}, s, t, M) = \text{True}$

现在我们说明算法 13.1 的系统的确是一个签名方案。

如果签名的验证是 True, 就表明 Alice 拥有  $\text{ID} \cdot t^{h(t \parallel M)}$  模  $N$  惟一确定的  $e$  次方根 ( $\text{ID} \cdot t^{h(t \parallel M)}$  模  $N$  的  $e$  次方根就是  $s$ , 惟一性由  $\gcd(e, \phi(N)) = 1$  保证)。

$\text{ID} \cdot t^{h(t \parallel M)}$  的构造不是一件困难的事。例如, 可以选择一个随机数  $r$ , 构造  $h(t \parallel M)$ , 然后计算  $t^{h(t \parallel M)} \pmod{N}$  并乘以身份 ID, 最后得到结果  $\text{ID} \cdot t^{h(t \parallel M)}$ 。然而, 因为在构造过程中引入了密码杂凑函数, 所构造的值是可识别的, 但要求所构造的值的  $e$  次方根是困难的。因此, 假设 Alice 拥有身份 ID 的  $e$  次方根, 这是 Trent 发给 Alice 的私钥, 她能够使用该密钥构造一个签名对。

然而, 我们没有对 Shamir 的基于身份的签名方案的不可伪造性给出正式且强有力的证明。这是因为伪造签名的困难性与构造  $\text{ID} \cdot t^{h(t \parallel M)} \pmod{N}$  和求解  $\text{ID} \cdot t^{h(t \parallel M)}$  的  $e$  次方根的困难性相关, 它的困难性必定与所用的杂凑函数的细节 (另外还有 RSA 问题) 有关。与其他数字签名方案的安全性证明类似, 对基于 Shamir 的身份的签名方案安全性严格的证明需要杂凑函数性能的一个形式模型。这样的模型将在以后的章节介绍。

### 13.3.2 基于 ID 的密码确切提供了什么

在通常意义下的公钥密码中, Bob 运用 Alice 的公钥来验证 Alice 的签名, 同时他也应该验证 Alice 公钥的真实性。例如通过 Bob 可以验证 Alice 的公钥证书 (Alice 的公钥与她身份相联) 来验证 Alice 公钥的真实性。即 Bob 应该确信与 Alice 的密钥信道已经正确地建立 (见图 10.1)。

在基于 ID 的签名方案中,意识到 Bob 不需要执行一个密钥信道的建立的认证是十分重要的。当 Bob 验证签名为 True 时,同时表明以下两个问题:

- Alice 用基于她的 ID 的私钥生成签名。
- Alice 的 ID 已经被 Trent 认证,她的 ID 证书使得 Alice 可以生成签名。

在一个逻辑步骤内能够同时验证两件事是基于 ID 签名方案所提供的一个很好的特征。能够避免从签名者到验证者的证书传递,节约通信带宽。这种特性给基于 ID 的密码体制带来另一个名字:非交互式的公钥密码体制。不久我们将看到非交互式的公钥密码体制在基于 ID 的加密系统中有更重要的意义。

最后我们必须重述并记住 Trent 能够伪造用户的签名!因此,Shamir 的基 ID 的签名方案不适于开放系统环境中的应用。但适于封闭式系统环境中的应用。在封闭式系统环境中,Trent 对整个系统中所有信息拥有合法的所有权。遗憾的是,这个限制太强了。

一个具有挑战性的有待解决的问题是设计一个没有上面那种限制的基于 ID 的签名方案。另一个有待解决的问题是设计一个基于 ID 的可以非交互吊销密钥的签名方案。当用户的私钥被泄露后,就必须吊销密钥。

如果以上两个公开问题得不到解决,基于 ID 的签名方案的应用就将受到很大的限制。在本章的其余部分我们将看到,上面两个问题中的一个在基于身份 ID 的加密方案中被解决,不再需要对 Trent 的绝对信任。

### 13.3.3 自证实公钥

假设 $(s, P)$ 是一公、私钥对,公钥认证框架提供了一个密钥对和一种保证书  $G$ ,它将  $P$  与身份  $I$  联系起来。

在一个基于目录的公钥认证框架中(如例 13.1 中的 X.509),保证书  $G$  上有 CA 对 $(I, P)$ 的数字签名。这个认证框架由四个不同的属性值 $(s, I, P, G)$ 构成。其中的三个属性值 $(I, P, G)$ 是公开的,且可以在公共目录上获得。当一个主体需要  $I$  的公钥时,他可以得到公开的三元组 $(I, P, G)$ 。用 CA 的公钥来验证  $G$ ,之后利用  $P$  来认证用户。

在基于身份的认证框架中(如 13.3.1 节中 Shamir 的方案),公钥就是身份  $I$ 。因此, $P = I$ ,并且这个认证框架由两个属性值 $(s, I)$ 构成。正如我们在 13.3.1 节中所看到的,当一个主体需要认证 Alice 的公钥  $I$  时,就必须验证他的签名。结果为 True 表明这个公钥是真实的。所以保证书不是别的什么,而就是他的私钥,即  $G = s$ 。

Girault 建议一种公钥认证框架方案,它介于基于证书方案和基于身份方案之间[124, 123]。在 Girault 方案中,保证书等于公钥,即  $G = P$ ,所以可以说它是自证实的,每个用户都有三个属性 $(s, P, I)$ 。在 Girault 方案中,用户的私钥可以由用户选择。

#### 13.3.3.1 Girault 方案

Girault 方案仍然需要一个可信赖的机构 TA(假设它是 Trent),TA 建立系统参数,并帮助用户建立她/他的密钥属性。

#### 13.3.3.2 系统密钥数据

Trent 生成 RSA 密钥数据如下:



1. 一个公开模数  $N = PQ$ , 其中  $P, Q$  是长度相等大素数, 例如  $|P| = |Q| = 512$ ;
2. 一个公开指数  $e$  且与  $\phi(N)$  互素, 其中  $\phi(N) = (P-1)(Q-1)$ ;
3. 一个秘密指数  $d$  且满足  $ed \equiv 1 \pmod{\phi(N)}$ ;
4. 一个公开元素  $g \in \mathbb{Z}_N^*$  具有最大的乘法阶; 为了计算  $g$ , Trent 找  $g_P$  作为模  $P$  的生成元, 并找  $g_Q$  作为模  $Q$  的生成元, 然后 Trent 可以运用中国剩余定理来构造  $g$  (参见 6.2.3 节的定理 6.7)。

Trent 公开系统参数  $(N, e, g)$ , 并秘密保存系统私钥  $d$ 。

### 13.3.3.3 用户的密钥数据

Alice 随机选择一个长度为 160 比特的整数  $s_A$  作为私钥, 计算

$$v \leftarrow g^{-s_A} \pmod{N}$$

并把  $v$  发送给 Trent。然后, 她运用在 13.3.3.4 节中所描述一个简单协议向 Trent 证明她知道  $s_A$  且不泄漏  $s_A$ , Alice 也发送她的身份  $I_A$  给 Trent。

Trent 创建 Alice 的公钥为  $v - I_A$  的 RSA 签名:

$$P_A \leftarrow (v - I_A)^d \pmod{N}$$

Trent 发送  $P_A$  给 Alice 作为 Alice 公钥的一部分。因此, 下面的等式成立:

$$I_A \equiv P_A^e - v \pmod{N} \quad (13.3.2)$$

表面看来, 在密钥的建立过程中, 由于  $P_A$  和  $v$  是  $\mathbb{Z}_N^*$  两个随机数, 因此看起来构造式 (13.3.2) 似乎不困难。例如, Alice 随机选取  $P_A$  并根据式 (13.3.2) 用  $P_A^e$  和  $I_A$  计算  $v$ 。然而, 如果按这种方式来计算  $v$ , Alice 就不能知道它以  $g$  为底模  $N$  的离散对数。

Alice 能够证明她知道以  $g$  为底模  $N$  的离散对数, 即值  $-s_A$ , 这就保证了  $P_A$  是由 Trent 发行的。完成这个证明的最简单方法是运用将在 13.3.3.4 节中介绍的 Diffie-Hellman 密钥交换协议的一个变形。

### 13.3.3.4 密钥交换协议

假设  $(s_A, P_A, I_A)$  是 Alice 的公钥数据,  $(s_B, P_B, I_B)$  是 Bob 的公钥数据。他们可以通过协商简单地交换一个认证的密钥:

$$K_{AB} \equiv (P_A^e + I_A)^{s_B} \equiv (P_B^e + I_B)^{s_A} \equiv g^{-s_A s_B} \pmod{N}$$

在这个密钥协商中, Alice 计算  $(P_B^e + I_B)^{s_A} \pmod{N}$ , Bob 计算  $(P_A^e + I_A)^{s_B} \pmod{N}$ 。因此, 这的确是一个 Diffie-Hellman 密钥协商协议。如果双方能够协商相同的密钥, 那么他们就知道另一方已经证明了她/他的身份。

Girault 也提出了一个基于 ID 的识别协议, 同时也在 ElGamal 签名方案中给出了一个基于身份的签名方案 [124]。

### 13.3.3.5 讨论

Girault 的自证实公钥拥有 Shamir 的基于身份的公钥的一个特点: 不需要对可信第三方发给密钥所有者的密钥证书进行验证。这个验证是暗含的, 并且与验证密钥所有者的密码能力同时进行。



但是,验证者除了需要一个身份的同时还需要一个独立的公钥,即除了  $I$  外还需要  $P$ ,前者不能由后者得到,这就意味着验证者在使用密钥所有者的公钥之前,必须向其发出使用公钥的请求。这是一个额外的通信步骤。因此, Girault 的自证实公钥不是非交互的公钥密码体制(见 13.3.2 节的讨论),这是自证实公钥的一个缺陷。

### 13.3.4 利用“弱”椭圆曲线对构造基于身份的公钥密码体制

Shamir 最初的基于身份的公钥密码体制是一个数字签名方案。他也猜想基于身份的加密系统是存在的。1984 年 Shamir 提出基于身份的密码系统以后,几个不同的基于身份的密码体制也被提出[253, 52, 79, 143, 193, 291, 289]。

Sakai、Ohgishi 与 Kasahara[253]和 Joux[156]分别独立地提出了利用对映射函数的一种特殊性质的思想(见 5.5 节),这个映射函数作用于椭圆曲线上的点所构成的阿贝尔群上。Sakai、Ohgishi 与 Kasahara[253]的工作是前面所述分析结果(在 13.3.4.1 节中解释)的一种奇妙应用,此分析结果使 Shamir 的猜想变成现实。Joux[156]的工作独立地使用了同样的技术来完成另一种应用:一轮三方 Diffie-Hellman 密钥分享, Joux 称其为“三方 Diffie-Hellman”协议。

Sakai 等[253]和 Joux[156]独立地应用了组对技术,不仅完成了以前不知道如何去做的事情,更重要的是,他们把以前 Menezes、Okamoto 和 Vanstone 的密码分析结果[199](关于椭圆曲线的一个负面结果)转化为积极的应用。他们在学术上的贡献于 2000 年以后再一次引起了人们对基于对身份的密码的兴趣。

Sakai 等[253]和 Joux[156]所使用的一条特殊性质如下:在对映射可以有效计算的“弱”椭圆群中,判定 Diffie-Hellman 问题(DDH 问题)是容易的,然而计算 Diffie-Hellman 问题(CDH 问题)是困难的。先让我们研究下一类弱的椭圆曲线情况以及相对容易的 DDH 问题。在这个研究之后,我们将介绍基于对的密钥协商方案。

#### 13.3.4.1 一类“弱”的椭圆曲线

Menezes、Okamoto 和 Vanstone[199]给出了定义在有限域上的一类特殊的椭圆曲线,存在一个有效算法,将曲线(在有限域上)上的两点映射到基域中的一个元素。这类特殊的曲线被称为超奇异曲线,并满足在式(5.5.6)中的  $t$  能够被基域的特征整除。像在 5.5.6 节一样,我们只讨论域特征大于 3 的简单情况。

假设  $E(\mathbb{F}_q)$  是一条曲线( $q$  是一个素数幂)。对于某个大素数  $\alpha \nmid \#E(\mathbb{F}_q)$  ( $\alpha, q$  互素),由拉格朗日定理(推论 5.2),我们知道群  $E(\mathbb{F}_q)$  包含着许多阶数为  $\alpha$  的点(这些点  $P$  满足  $[\alpha]P = \mathcal{O}$ , 如果必要,请回顾 5.5 节)。此外,群  $E(\mathbb{F}_{q^\ell})$  是非循环群且包含不相交的阶为  $\alpha$  的点的子群(注意,基域是  $\mathbb{F}_q$  的一个扩域  $\mathbb{F}_{q^\ell}$ , 其中  $\ell$  是某个整数,即一个曲线上点的坐标  $(x, y)$  是  $\mathbb{F}_{q^\ell}$  上的元素,如果必要,请参考 5.4 节中的扩域概念)。也就是说,在群  $E(\mathbb{F}_{q^\ell})$  中存在阶为  $\alpha$  的点  $P, Q$  满足  $P \notin \langle Q \rangle$  且  $Q \notin \langle P \rangle$ 。因为对于所有的整数  $v, u$ , 这些点满足  $P \neq [u]Q$  和  $Q \neq [v]P$ , 也就是说它们是线性独立的。两个线性无关阶为  $\alpha$  的点形成一个产生群  $E(\mathbb{F}_{q^\ell})$  上所有阶为点  $\alpha$  的一组基。这点可以证明。

对于这个素数  $\alpha$ , 扩域  $\mathbb{F}_{q^\ell}$  作为一个非零元素的乘群也是一个阶为  $\alpha$  的子群,注意这个子群是惟一的(回顾定理 5.11,  $\mathbb{F}_{q^\ell}$  中的非零元素组成的乘群是循环群,且定理 5.2(2)说明其惟一性)。

众所周知,  $E(\mathbb{F}_{q^\ell})$  曲线中的所有  $\alpha$  阶点和  $\mathbb{F}_{q^\ell}$  中的  $\alpha$  阶子群之间存在一一且保持运算的映射。**Weil** 对就可以提供如此的映射。Menezes、Okamoto 和 Vanstone[199]证明对于超奇异椭圆曲线  $E(\mathbb{F}_q)$ , 在  $\ell \leq 6$  的小扩域上能够构造 Weil 对映射。这个小域扩张是非常重要的, 很快我们将回到这点上。

Weil 对把  $E(\mathbb{F}_{q^\ell})$  上阶数为  $\alpha$  的两个点映射到阶数为  $\alpha$  的子群  $\mathbb{F}_{q^\ell}$  中的一个元素。假设  $P, Q$  是子群  $E(\mathbb{F}_{q^\ell})$  上阶为  $\alpha$  的两个点。我们表示 Weil 对为

$$e_\alpha(P, Q)$$

在这个表示中, 下标  $\alpha$  指定了  $P, Q$  是  $\alpha$  阶子群中的点。注意这些点可能在同一个子群中(包括  $P = \mathcal{O}$  或  $Q = \mathcal{O}$ ), 这时它们是线性相关的, 当它们在不同的  $\alpha$  子群中时, 则线性无关。

Weil 对满足以下性质。

**性质 13.1 Weil 对性质** 假设  $\alpha$  是一个素数,  $P, Q, R$  是  $E(\mathbb{F}_{q^\ell})$  中阶为  $\alpha$  的子群中的点。

Weil 对  $e_\alpha$  具有以下性质:

**单位元:** 对于所有的  $P$ :

$$e_\alpha(P, P) = 1$$

**双线性性:** 对于所有的  $P, Q, R$ :

$$e_\alpha(P + Q, R) = e_\alpha(P, R)e_\alpha(Q, R), e_\alpha(R, P + Q) = e_\alpha(R, P)e_\alpha(R, Q)$$

**非退化性:** 对于所有的  $P, Q$  线性无关(它们在不同的  $\alpha$  阶子群中):

$$e_\alpha(P, Q) \neq 1, e_\alpha(Q, P) \neq 1$$

**实效性:** 对于所有的  $P, Q$ ,  $e_\alpha(P, Q)$  和  $e_\alpha(Q, P)$  是实际有效可计算的。

注意, 由双线性性我们有

$$e_\alpha([n]P, Q) = e_\alpha(P, Q)^n = e_\alpha(P, [n]Q)$$

进而, 由非退化性我们知道, 只要  $P, Q$  线性无关且  $\alpha \nmid n$ , 这个“对映射”的结果不是  $\mathbb{F}_{q^\ell}$  中的乘法单位, 因此, 该对映射不是无意义的。由于在非退化情况下, 映射结果是素数阶  $\alpha$  中的一个非单位元, 因此, 它是  $E(\mathbb{F}_{q^\ell})$  中一个单位的次根。

从椭圆曲线上离散对数的困难问题(ECDLP, 在 5.5.3 节中定义)到有限域上离散对数的困难性问题, Weil 对的这些特性能够提供意义深远的、被称为 **MOV 归约**(MOV reduction)的归约。在已知阶为  $\alpha$  点对  $(P, [n]P)$  的情况下, 为了应用 MOV 归约到 ECDLP, 对某个与  $P$  线性无关的点, 我们可以计算 Weil 对  $\xi = e_\alpha(P, X)$  和  $\eta = e_\alpha([n]P, X)$ 。注意,

$$\eta = e_\alpha([n]P, X) = e_\alpha(P, X)^n = \xi^n \quad (13.3.3)$$

因为  $P, Q$  线性无关, 通过非退化  $e_\alpha(P, X) \neq 1$ , 因此在  $\mathbb{F}_{q^\ell}$  中的对  $(\xi, \eta)$  提供了有限域中离散对数问题。我们知道对于后者, 有一个时间复杂度表示为  $\text{sub\_exp}(q^\ell)$  的亚指数时间解法[在复杂度表达式(8.4.2)中用  $q^\ell$  代替  $q$ ]。回忆我们早期所讨论的  $\ell \leq 6$  时的超奇异曲线。因此, MOV 归约是一个强有力的方法: 它把普遍认为的指数复杂度问题  $O(\sqrt{\alpha}) \approx O(\sqrt{q})$  归结为一个  $\ell$  不超过 6 的亚指数问题  $\text{sub\_exp}(q^\ell)$ , 通常  $\ell = 2$ 。

考虑到硬件计算技术的进步导致  $q$  的长度可以增大, 一个超奇异曲线的参数也必须根据有限域的变化而变化。换句话说, 在密码中使用椭圆曲线的优点将会消失。因此, 在 Menezes、

Okamoto 和 Vanstone[199]的密码分析工作后,把超椭圆曲线排除在密码应用之外成了一个普遍接收的约定。它们是弱曲线。

那么在这一小节的标题中我们为什么用“弱”这个词呢?这些曲线有一个新发现的有用特性,该特性对研究团体具有强大的震撼力。首先让我们看一看 Diffie-Hellman 问题的判定形式。

#### 13.3.4.2 判定 Diffie-Hellman 问题

在定义 8.1 见(8.4 节)中,我们介绍了 CDH 问题(Diffie-Hellman 问题的计算性版本)。Diffie-Hellman 问题的判定形式在定义 13.1 中给出。

##### 定义 13.1 判定 Diffie-Hellman 问题(DDH 问题)

输入 desc( $G$ ):一个阿贝尔群  $G$  的描述;  
 $(g, g^a, g^b, g^c) \in G^4$ , 其中  $g$  为群  $G$  的生成元;  
 输出 YES if  $ab \equiv c \pmod{\#G}$ 。

DDH 问题可能不比 CDH 问题困难。如果存在着一个 CDH 问题的解法(这样假设的解法通常称为预言机),那么当输入  $(g, g^a, g^b, g^c)$  时,预言机能够由输入的前三个元素求得  $g^{ab}$ ,因此,能够回答 DDH 问题通过检验 CDH 预言机的输出是否等于  $g^c$ 。

然而,在通常的阿贝尔群中,我们并不确切地知道关于这两个问题之间关系更多的东西。此外,我们不知道关于求解 DDH 问题的有效算法。回答 DDH 问题的困难性引出求解一个普遍认为困难的问题(将在假设 14.2 中描述),这个假设是许多密码系统安全的前提条件,例如 [21, 59, 85, 211, 285]。

对于超奇异椭圆曲线的特殊情况,我们知道最近发现的以下事实:DDH 问题是容易的。这点被 Joux 和 Nguyen[157]证实。在解释这个事实之前,我们需要把问题(CDH 问题, DH 问题)转化为加法形式,因为椭圆曲线群是以加法形式描述的。

##### ( $G, +$ ) 中的离散对数(DL)问题

输入 两个元素  $P, Q \in G$ , 其中  $P$  为群的生成元;  
 输出 一个整数  $a$ , 满足  $Q = aP$ 。

##### ( $G, +$ ) 中的计算 Diffie-Hellman(CDH)问题

输入 三个元素  $P, aP, bP \in G$ , 其中  $P$  为群的生成元;  
 输出 元素  $(ab)P \in G$ 。

##### ( $G, +$ ) 中的判定 Diffie-Hellman(CDH)问题

输入 四个元素  $P, aP, bP, cP \in G$ , 且  $P$  是一个生成元;  
 输出 YES 当且仅当  $c \equiv ab \pmod{\#G}$ 。

#### 13.3.4.3 “弱”曲线使判定 Diffie-Hellman 问题容易

Weil 对的单位元性质(见性质 13.1)有时是难以实施的,这就意味着对于群  $G_1$  中的  $P, Q$  (对于某个数  $a$  满足  $Q = [a]P$ ), 对  $e_a(P, Q) = e_a(P, P)^a = 1^a = 1$ 。为了获得一个非退化的映射结果,必须找到另一个阶为  $\alpha$  的点  $X$ , 且  $X$  与  $G_1$  中的元素线性独立。这在很大程度上限制了 Weil 对在密码应用中的积极作用。

Verheul[298]设计了一个他命名为“变形映射”的方法。一个变形映射就是对曲线上点的坐标修正(这种修正在基域  $\mathbb{F}_q$  上进行)。假设  $\Phi(P)$  表示这种修正。对于  $E(\mathbb{F}_{q^t})$  中的一个点  $P$ ,

只要  $P$  的阶大于 3, 那么  $\Phi(P)$  也是  $E(\mathbb{F}_{q^\ell})$  中一个具有同样阶的点。通常的处理方法是在“基域”  $\mathbb{F}_q$  中进行修正。也就是说, 选择一个  $P \in E(\mathbb{F}_q)$  并修正它的坐标, 以使  $\Phi(P) \in E(\mathbb{F}_{q^\ell})$ , 这种通常的处理保证了  $P$  和  $\Phi(P)$  是线性无关的。这是因为现在  $P$  的坐标和  $\Phi(P)$  的坐标  $(x, y)$  在不同的域中, 然而点乘  $[u]P, [u]\Phi(P)$  从来不在同一个交域上 (见 5.5.2 节, 即对所有的整数  $u, v, [u]P \in E(\mathbb{F}_q), [v]\Phi(P) \in E(\mathbb{F}_{q^\ell})$ )。

在这个变形映射下, Weil 对被修正为

$$e(P, P) = e_a(P, \Phi(P))$$

现在很清楚, 由于  $P$  和  $\Phi(P)$  是线性独立的, 所以  $e(P, P) \neq 1$ 。此外, 对于  $P, Q \in G_1$ , 我们修正的 Weil 对具有以下对称性质:

$$e(P, Q) = e(P, [n]P) = e(P, P)^n = e([n]P, P) = e(Q, P) \quad (13.3.4)$$

令  $G_1$  表示  $E(\mathbb{F}_{q^\ell})$  的一个  $\alpha$  阶子群 (事实上, 为简明起见, 总有  $G_1 \subset E(\mathbb{F}_q)$ ), 令  $G_2$  表示  $\mathbb{F}_{q^\ell}$  的一个  $\alpha$  阶子群。由 Weil 对的双线性给出的操作保留属性, 我们知道修改过的 Weil 对实际上是  $G_1$  和  $G_2$  间的一个同型。

现在我们考虑  $G_1$  中的 DDH 问题。为了回答一个四元组  $(P, [a]P, [b]P, [c]P)$  是否是一个 DH 四元组, 我们计算  $\xi = e(P, [c]P)$  和  $\eta = e([a]P, [b]P)$ 。由于  $\xi = e(P, P)^c$ ,  $\eta = e(P, P)^{ab}$  且  $e(P, P) \neq 1_{G_2}$ , 四元组是一个 DH 四元组, 当且仅当  $\xi = \eta$ , 即当且仅当  $ab \equiv c \pmod{\#G_1}$ 。就改变的 Weil 对的实际效率来看, 可以有效地回答判定性问题。

Joux 和 Nguyen 的发现使超奇异椭圆曲线有了许多新的意义的密码应用。基于 ID 的密码是其中一个突出的应用。这些新的应用基于一个事实和迄今为止我们所讨论的一个困难性假设, 归纳如下:

#### Fact (DDH 问题是容易的)

在超奇异椭圆曲线群中, 通过 Weil 对算法可以有效地解答 DDH 问题。

#### Assumption (CDH、DL 问题是困难的)

在超奇异椭圆曲线群中, 通过适当地选取超奇异椭圆曲线的规模, DH 问题仍是困难的。对于曲线  $E(\mathbb{F}_q)$  而言, 它的困难性可表示  $\text{sub\_exp}(q^\ell)$ , 其中  $\ell \leq 6$ 。

在 CDH 问题、DL 问题仍然困难的假设下, 由于 MOV 归约和求解有限域上离散对数的亚指数算法的影响, 复杂度的表达式是  $\text{sub\_exp}(q^\ell)$ 。因此, 与通常的曲线相比现在我们必须增大超奇异曲线的安全参数 (曲线的大小)。这个加强的安全参数应该满足  $\text{sub\_exp}(q^\ell)$  是一个不可行的量。因此, 为了利用新发现的超奇异椭圆曲线的数学特性 (积极的应用将在 13.3.5 节至 13.3.7 节中给出), 我们选择了以牺牲效率为代价, 而效率是有限域上椭圆曲线的基本优势。

有两种对技术: 我们所讨论的修正的 Weil 对和 Tate 对, 后者更有效。对算法的详细讨论超出了本书的范围。有兴趣的读者可以参阅 [52, 118]。本章的其余部分将一直使用修改的 Weil 对。

### 13.3.5 Sakai、Ohgishi 和 Kasahara 的基于 ID 的非交互密钥分享系统

像 Shamir 的基于 ID 的签名方案一样, Sakai、Ohgishi 和 Kasahara [253] 的基于 ID 的密钥分享系统 (SOK 密钥分享系统) 也需要一个可信的机构 TA (称其为 Trent) 来操作密钥建立中心。

这个 SOK 密钥分享系统包含下面三个组成部分:

- **系统参数建立:** Trent 运行这个算法来建立全局系统参数和主密钥;
- **用户密钥生成:** Trent 运行这个算法;输入主密钥和一个任意比特串  $id \in \{0,1\}^*$ , 输出相应于  $id$  的私钥, 这是式(13.3.1)的一个实例;
- **密钥分享方案:** 两个端用户以非交互的方式执行该方案, 该方案以用户的私钥和意定通信方的公钥( $id$ )为输入;最后, 该方案输出一个由这两个用户共享的密钥。

这三个组成部分可通过以下步骤实现。

### 系统参数建立

在开放密钥生成中心为用户生成服务之前, Trent 首先建立系统参数。在系统参数的生成过程中, Trent 执行如下:

1. 生成阶数为素数  $p$  的两个群  $(G_1, +)$  和  $(G_2, \cdot)$ , 同时也生成修正的 Weil 对<sup>①</sup>  $e: (G_1, +)^2 \rightarrow (G_2, \cdot)$ 。任意选取一个生成元  $P \in G_1$ ;
2. 选取  $\ell \in {}_U\mathbb{Z}_p$ , 令  $P_{pub} \leftarrow [\ell]P$ ;  $\ell$  作为主密钥。
3. 选择一个强密码杂凑函数  $f: \{0,1\}^* \mapsto G_1$ , 该杂凑函数把用户的  $id$  映射到  $G_1$  中一个元素。

Trent 公布系统参数及对它们的描述:

$$(\text{desc}(G_1), \text{desc}(G_2), e, P, P_{pub}, f)$$

把  $\ell$  作为系统的密钥保存。由于 Trent 是整个系统都知道的主体, 系统中的所有用户都知道这些公开的系统参数(例如这些参数可能被固化到使用本方案的每个应用中)。

注意, 主密钥  $\ell$  的秘密性由  $G_1$  上 DLP 的困难性保证。

现在 Trent 开放密钥生成中心。

### 用户密钥生成

假设  $ID_A$  表示 Alice 的惟一可识别的身份。我们假设  $ID_A$  包含足够多的冗余以至于系统中其他用户不可能以  $ID_A$  作为她/他的身份。在对 Alice 的身份进行物理识别并确定  $ID_A$  的惟一性之后。Trent 的密钥生成服务如下:

1. 计算  $P_{ID_A} \leftarrow f(ID_A)$ , 这是  $G_1$  中的一个元素, 并且是 Alice 的基于 ID 的公钥;
2. Alice 的私钥为  $S_{ID_A}$ , 且满足  $S_{ID_A} \leftarrow [\ell]P_{ID_A}$ 。

应该注意经过杂凑函数作用以后,  $P_{ID_A}$  看起来是随机的。然而, 包含足够多可识别(冗余)信息且在密码杂凑函数  $f$  作用下的  $P_{ID_A}$  的原像  $ID_A$  是一个可识别的元素。因此, 把  $ID_A$  看做是 Alice 的公钥或者把  $P_{ID_A}$  看做是 Alice 的公钥, 在本质上是没有任何区别的。

我们还应该注意 Alice 的私钥由  $G_1$  中的 CDH 问题的困难性来保证, 这是因为  $P_{ID_A}$  一定是由  $P$  ( $P$  是  $G_1$  的生成元)生成的。因此, 我们可以用某个  $a < p$  来表示  $P_{ID_A} = [a]P$ , 那么由  $P$ ,  $P_{pub} (= [\ell]P)$ ,  $P_{ID_A} (= [a]P)$ , 求

<sup>①</sup> 最初的 SOK 密钥分享系统使用未修正的 Weil 对, 没有现在给出的方案方便。

$$S_{ID_A} = [\ell] P_{ID_A} = [\ell a] P$$

显然是  $G_1$  中的一个 CDH 问题。

### 密钥分享方案

对于用户 Alice 和 Bob 而言,  $ID_A$  和  $ID_B$  分别是他们的身份信息且他们相互知道对方的身份。因此,各自的公钥分别为  $P_A = f(ID_A)$  和  $P_B = f(ID_B)$ , 而且他们彼此也知道。

Alice 通过计算

$$K_{AB} \leftarrow e(S_{ID_A}, P_{ID_B})$$

可以产生一个分享的密钥  $K_{AB} \in (G_2, \cdot)$ 。

Bob 通过计算

$$K_{BA} \leftarrow e(S_{ID_B}, P_{ID_A})$$

可以产生一个分享的密钥  $K_{BA} \in (G_2, \cdot)$ 。

注意根据这个对的双线性特性(性质 13.1), 我们可以得到

$$K_{AB} = e(S_{ID_A}, P_{ID_B}) = e([\ell] P_{ID_A}, P_{ID_B}) = e(P_{ID_A}, P_{ID_B})^\ell$$

同理,

$$K_{BA} = e(P_{ID_B}, P_{ID_A})^\ell$$

由修正 Weil 对的对称性(13.3.4), 我们有

$$K_{AB} = K_{BA}$$

因此,即使 Alice 和 Bob 不交互信息,他们也确实能够分享一个密钥。

对于除 Alice、Bob 和 Trent 之外的另一方而言,从公共数据  $(P, P_{ID_A}, P_{ID_B}, P_{pub})$  求  $K_{AB}$  是一个 **双线性 Diffie-Hellman 问题**[52], 它本质上是一个 CDH 问题。

当 Bob 收到一条用  $K_{AB}$  认证的消息时,只要这条消息不是他本人发送的,他就确切地知道 Alice 是这条消息的所有者。然而,因为 Bob 同样具有构建这个消息的密码能力,尽管 Alice 向指定验证者 Bob 证明了消息的来源,她仍然可以在第三方面前否认他参与了通信。可以考虑 Alice 和 Bob 是间谍时的情况,当他们联系时,他们必须向对方认证自己。然而,作为一个双重代理, Alice 可能担心 Bob 也是一个双重代理。因此,一个对间谍的认证方案必须有一个绝对可否认的认证特性。SOK 密码分享系统恰好具有这样的特性。它是一个基于公钥的系统,即认证不需要基于在线的可信第三方(如同在第 2 章中介绍的基于分享秘密的认证一样)。

对于 SOK 密钥分享系统的一个重要的应用是 Internet 密钥交换协议(前面章节所介绍的 IKE)。IKE 协议具有看似合理的否认能力的认证方式(见 12.2.4 节)。显然, Sakai 等人的密钥分享系统的完全否认特性提供了一个更好的解决方案,而这个协议仍然是一个基于公钥的协议。

### 13.3.6 三方 Diffie-Hellman 密钥协商

Joux[156]应用对技术以一种出奇简单的方法完成了三方间的密钥协商。他把他的协议命名为“三方 Diffie-Hellman”。Joux 的最初方案使用了 Weil 对,因此,它不太适合实际应用(必须构造线性独立的点)。我们将以修正 Weil 对来引入这个方法。



假设 Alice 通过计算

$$P_A \leftarrow [a]P$$

构造她的密钥协商成分  $P_A$ , 其中  $P \in G_1$  是超奇异椭圆曲线上一个阶数为  $\alpha$  (素数) 的点, 且  $a < \alpha$  是一个整数。同样, 假设 Bob 和 Charlie 的密钥协商成分分别是

$$P_B \leftarrow [b]P, P_C \leftarrow [c]P$$

其中  $b < \alpha, c < \alpha$ 。整数  $a, b, c$  分别是他们的私钥。

这三方交换  $P_A, P_B, P_C$ , 即他们在公开目录公布他们的密钥协商成分。完成之后, 他们分享下面的密钥:

$$e(P_B, P_C)^a = e(P_A, P_C)^b = e(P_A, P_B)^c = e(P, P)^{abc}$$

Alice 通过对第一个对进行求幂来计算分享的密钥, Bob 对第二个对、Charlie 对第三个对进行同样的操作。

如果不使用对技术, 三方的 Diffie-Hellman 密钥交换不可能通过简单的一轮计算得到。

当然, 正如在最初的 Diffie-Hellman 密钥交换协议中, 这个方案不具有认证特性。

### 13.3.7 Boneh 和 Franklin 的基于 ID 的密码体制

因为两个主体仅仅通过他们的身份就可以在它们之间建立一个分享的密钥, 加密也可以通过他们的身份来建立。Boneh 和 Franklin[52] 应用对技术建立了第一个实用的基于 ID 的公钥密码体制, 该密码体制完全满足 Shamir 所声称的基于 ID 的公钥密码系统。

Boneh 和 Franklin 的基于 ID 的密码体制由四个算法组成:

- **系统参数的建立** Trent 运行该算法来生成系统的全局参数和主密钥;
- **用户密钥的生成** Trent 运行该算法, 输入主密钥和一个任意的比特串  $id \in \{0, 1\}^*$ , 该算法输出相应于  $id$  的私钥, 这是式(13.3.1)的一个实例;
- **加密** 这是个概率算法, 用公钥  $id$  来加密消息;
- **解密** 把密文和私钥输入该算法, 最后返回相应的明文。

算法 13.2 详细说明了 Boneh 和 Franklin 的基于 ID 的密码体制。

#### 算法 13.2 Boneh 和 Franklin 的基于 ID 的密码体制

系统参数的建立(由 Trent 执行)

1. 生成两个阶数为素数  $p$  的两个群  $(G_1, +)$  和  $(G_2, \cdot)$ , 一个对映射  $e: (G_1, +)^2 \mapsto (G_2, \cdot)$ 。任意选择一个生成元  $P \in G_1$ 。
2. 选取  $s \in {}_U\mathbb{Z}_p$  并令  $P_{pub} \leftarrow [s]P$ ,  $s$  作为主密钥。
3. 选择一个强密码杂凑函数  $F: \{0, 1\}^* \mapsto G_1$ , 这个杂凑函数把用户的身份  $id$  映射到  $G_1$  中一个元素。
4. 选择一个强密码杂凑函数  $H: G_2 \mapsto \{0, 1\}^n$ , 这个杂凑函数决定  $\mathcal{M}$  (明文空间) 是  $\{0, 1\}^n$ 。

Trent 把  $s$  作为系统的私钥保存, 并公开系统参数和它们的描述

$$(G_1, G_2, e, n, P, P_{pub}, F, H)$$

### 用户密钥生成

假设 ID 表示用户 Alice 的惟一可识别的身份。对 Alice 进行物理鉴定以确信 ID 具有惟一性。Trent 的密钥生成如下：

1. 计算  $Q_{ID} \leftarrow F(ID)$ , 这是  $G_1$  中的一个元素, 并且也是 Alice 的基于身份的公钥。
2. Alice 的私钥为  $d_{ID}$ , 且满足  $d_{ID} \leftarrow [s]Q_{ID}$ 。

### 加密

为了发送秘密消息给 Alice, Bob 首先获得公开参数  $(G_1, G_2, e, n, P, P_{pub}, F, H)$ 。运用这些参数, Bob 计算

$$Q_{ID} = F(ID)$$

假设消息被分成  $n$  比特的块, 为了加密  $M \in \{0, 1\}^n$ , Bob 选取一个数  $r \in_U \mathbb{Z}_p$  并计算

$$\begin{aligned} g_{ID} &\leftarrow e(Q_{ID}, [r]P_{pub}) \in G_2 \\ C &\leftarrow ([r]P, M \oplus H(g_{ID})) \end{aligned}$$

所得的密文为  $C = ([r]P, M \oplus H(g_{ID}))$ 。

### 解密

假设  $C = (U, V) \in \mathcal{C}$  是用 Alice 的公钥 ID 加密的密文。为了用它的密钥  $d_{ID} \in G_1$  来解密  $C$ , Alice 计算

$$V \oplus H(e(d_{ID}, U))$$

现在我们证明算法 13.2 所描述的系统的的确是一个密码系统。我们注意到

$$e(d_{ID}, U) = e([s]Q_{ID}, [r]P) = e(Q_{ID}, [r]P)' = e(Q_{ID}, [rs]P) = e(Q_{ID}, [r]P_{pub}) = g_{ID}$$

因此, 在解密过程中, Alice 输入杂凑函数  $H$  的值实际上是  $g_{ID}$ , 也就说与 Bob 在加密过程中输入杂凑函数的值一样。又因为异或运算是自取逆, 则

$$V \oplus H(e(d_{ID}, U)) = M \oplus H(g_{ID}) \oplus H(g_{ID}) = M$$

Boneh 和 Franklin 也给出了对基于 ID 的加密方案的安全性的形式化证明。这个安全概念是很强的: 自适应选择密文攻击。在 ElGamal 这一类型的加密中, 杂凑函数的直接使用意味着证明基于所谓的“随机预言模型”。由于在 V 部分我们将研究形式化的强安全概念和随机预言模型, 在这里我们不介绍安全证明技术。

#### 13.3.7.1 一个开放式系统的扩展

我们必须注意到在系统中 Trent 能够解密发给每个主体的密文消息。因此, Boneh 和 Franklin 的基本方案不适合在开放式系统中应用。然而, 他们的基本方案可以扩展到适合开放式系统中的应用。在这里我们描述一个扩展的方法, 该方法是在 Boneh 和 Franklin 的文章中讨论的方法的一个简单变体。

它的基本思想是使用多个 TA。然而,只有在不引起单个用户 ID 的数量爆炸也不引起密文长度增加的情况下,这种做法才是有意义的。这里有一个实现方法,我们将描述两个 TA 的情况,并且很容易扩展到多个 TA 的情况。

**系统参数建立** 假设参数  $(G_1, G_2, e, n, P, F, H)$  的定义同于 13.3.7 节,进一步

$$\begin{aligned} P_1 &\leftarrow [s_1]P, \\ P_2 &\leftarrow [s_2]P \end{aligned}$$

三元组  $(P, P_1, P_2)$  起到了 13.3.7 节中  $(P, P_{pub})$  的作用,也就是说,  $s_1$  和  $s_2$  分别是  $TA_1$  和  $TA_2$  的主密钥。

因此,  $(G_1, G_2, e, n, P, P_1, P_2, F, H)$  是系统的公共参数,这些参数被固化到应用中。

**用户密钥的生成** 假设 ID 表示用户 Alice 惟一可识别的身份,对于  $i = 1, 2$ ,  $TA_i$  按如下方式生成密钥:

1. 计算  $Q_{ID} \leftarrow F(ID)$ , 这是  $G_1$  中的一个元素,同时它也是 Alice 惟一基于 ID 的公钥;
2. 设置 Alice 的私钥为  $d_{ID}^{(i)} \leftarrow [s_i]Q_{ID}$ 。

最后, Alice 的私钥是下面的和:

$$d_{ID} = d_{ID}^{(1)} + d_{ID}^{(2)}$$

如果这两个 TA 不勾结,那么他们就不知道这个私钥。

注意, Alice 有惟一的公钥 ID。

**加密** 为了发送一条秘密消息给 Alice, Bob 首先获得系统参数  $(G_1, G_2, e, n, P, P_1, P_2, F, H)$ 。然后,利用这些参数计算

$$Q_{ID} = F(ID)$$

假设消息被分成  $n$  比特的块,为了加密  $M \in \{0, 1\}^n$ , Bob 选取一个数  $r \in_U \mathbb{Z}_p$  并计算

$$\begin{aligned} g_{ID} &\leftarrow e(Q_{ID}, [r](P_1 + P_2)) \\ C &\leftarrow ([r]P, M \oplus H(g_{ID})) \end{aligned}$$

这个密文就是  $C$ 。因此,密文是由  $G_1$  中的一个点和  $\{0, 1\}^n$  中的一个比特串组成的对。即密文空间  $C \in G_1 \times \{0, 1\}^n$ 。

**解密** 假设  $C = (U, V) \in C$  是用 Alice 的公钥 ID 加密的密文,为了用她的私钥  $d_{ID} \in G_1$  解密密文  $C$ , Alice 计算

$$V \oplus H(e(d_{ID}, U))$$

注意

$$\begin{aligned} e(d_{ID}, U) &= e([s_1]Q_{ID} + [s_2]Q_{ID}, [r]P) \\ &= e([s_1]Q_{ID}, [r]P) e([s_2]Q_{ID}, [r]P) \\ &= e(Q_{ID}, [rs_1]P) e(Q_{ID}, [rs_2]P) \\ &= e(Q_{ID}, [r](P_1 + P_2)) \\ &= g_{ID} \end{aligned}$$

因此, Alice 恢复了  $g_D$ 。因为比特异或是自取逆运算, 于是 Alice 能够解密:

$$V \oplus H(e(d_D, U)) = M \oplus H(g_D) \oplus H(g_D) = M$$

#### 讨论

- 与一个 TA 时的情况相比, 加密和解密需要双倍的计算量, 但是 Alice 的 ID 数目没有增加, 密文的长度也没有增加。
- 勾结的 TA 能够联合起来解密, 但是他们中任何单个人都不能解密。当使用多个 TA 时, 对无勾结的信心就增大了。很容易看到, 增加 TA 的数目, Alice 的 ID 数目和密文的长度保持不变。然而, 加密和解密所需要的计算量随着 TA 的数目线性增加。
- 当使用几个 TA 时, 要对用户密文的解密需要所有的 TA 勾结。如果相信至少一个 TA 是可信赖的, 就可以防止 TA 的搭线窃听。因此, 这个扩展的 IBE 方案适合于开放环境下的应用。

### 13.3.8 非交互特性: 无密钥信道的认证

在通常意义下, 当 Bob 希望使用密码技术秘密向 Alice 发送一条消息时, 他应该首先在他与 Alice 之间建立一条密钥信道(见图 7.1)。在公钥密码下, 可能是一条基于目录的密钥信道, 例如基于发送者验证接收者的公钥证书。因此, 发送者应该首先向接收者发送一个获得她/他的公钥证书的请求。考虑到在开放式系统中, 主体无法记忆公钥与它的拥有者之间的关联。在发送用接收者的公钥加密的秘密消息之前, 为了在发送者和接收者之间建立一条密钥信道, 他们用有必要进行交互通信。

在基于 ID 的公钥密码体制中, 密钥信道的建立不仅是不必要的, 实际上也是不可能的, 认识到这一点是很有意义的。即使 Bob 向 Alice 请求发送她的基于 ID 的公钥, Bob 也无法验证接收到的基于 ID 的公钥是否的确来自 Alice。Bob 需要做的就是将获得的 ID 作为 Alice 的有效公钥, 并用它对消息进行加密。如果最后 Alice 能够解密 Bob 发送的密文, 那么就说明这个 ID 的确是 Alice 的公钥。因此, 只要每个 ID 是对个人惟一的精确描述(这一点必须由 TA 证实), 那么在用一个基于 ID 的加密算法发送一条秘密消息以前, 发送者就不需要与接收者交互了。这就是一个基于 ID 的公钥密码体制为什么称为非交互的公钥密码体制的缘故。

在 SOK 密钥分享系统下, 基于 ID 的密码方案的非交互特点是非常明显的(见 13.3.5 节)。一旦 Alice 和 Bob 注册了他们的基于 ID 的公钥以后, 甚至不需要任何通信, 他们就可以共享一条安全的密钥信道。这条分享的信道以两个密钥为基础。在 Alice 和 Bob 之间不需要执行任何协议, 就可以建立一条共享的安全信道。我们把这个非交互的密钥协商与使用 13.3.3.4 节的 Girault 的自证实公钥的密钥交换协议比较, 后者不是非交互的, 这是因为在他们建立一个共享密钥之前, 两个主体必须交换他们的公钥。

不可能直接证实一个公钥是否属于某个主体, 或者说缺少可公开验证的密钥证书, 使得基于 ID 的公钥密码(“间谍问题”)有重要的应用。在引入零知识证明协议以后, 我们将在以后的章节看到这种应用。

### 13.3.9 基于身份的公钥密码学的两个公开问题

首先, 让我们回顾一下用户的密钥生成过程

$$\text{私钥} = F(\text{主密钥}, \text{公钥})$$

在这个用户私钥的提取方法中,用户提交他所选择的公钥。这是个威胁的模型,用户可能就是潜在的恶意的,然而 TA 必须无条件地计算并把私钥返回给用户。

注意,为了使密码系统是一个基于 ID 的密码系统,或者说是没有交互的或无证书的密码系统,函数  $F$  必须是确定性的。这样,用户的密钥生成过程就不含随机输入。换句话说,每个用户的私钥是主密钥的一个确定的像。通常认为这个计算(对抵制主密钥的密码分析而言)是一个具有潜在不安全性的运算。在 Goldwasser-Micali 对 Diffie-Hellman 的确定性陷门函数模型批评之后[127],可以很容易地理解这个不全性。在标准的公钥密码应用中,TA 通过增加随机输入广泛地避免了这种不安全性。

研究具有随机私钥的基于 ID 的公钥密码是一件有意义的事,这是第一个公开问题。

第二个也是具有挑战性的问题,就是设计一个基于 ID 的具有非交互身份吊销特点的密码系统。如果用户的私钥被泄露,有必要撤消他的身份。

## 13.4 本章小结

本章我们介绍了几种实现公钥密码认证框架的技术,这些技术包括:使用分层组织证书机构的基于目录的证书框架,基于信任网络的非层次认证框架,以及公钥是非随机的和自证实的基于身份的认证框架。

最近基于身份的公钥密码的进步不仅提供了实际和简易的认证公钥方法,而且开辟了许多新的认证服务种类:无需证书的基于公钥的认证。一个不用证书的认证框架有许多有用的特点。在以后的章节中,我们将看到无证书认证服务的一个很好的应用。

## 习题

- 13.1 由式(13.1.1)中的私钥生成的公钥为什么必须进行证实?
- 13.2 对于 RSA、Rabin 和 ElGamal 公钥密码体制,式(13.1.1)中的  $F$  分别表示什么?
- 13.3 在公钥证书吊销之前的数字签名是否是无效的?
- 13.4 如果公钥像式(13.3.1)那样由私钥生成,为什么不需要证实公钥?
- 13.5 一个基于 ID 的密码体制为什么也称为非交互的公钥密码体制?
- 13.6 Girault 的自证实公钥密码体制是非交互的吗?
- 13.7 在 Girault 的自证实公钥密码体制的密钥交换协议(见 13.3.3.4 节)中,协议的参与者为什么能够否认参与协议的执行?
- 13.8 在密码的使用中,为什么超奇异椭圆曲线比非超奇异椭圆曲线需要更强的安全参数?
- 13.9 与 Diffie-Hellman 密钥协商协议(协议 8.1)不同,Girault 的自证实公钥密码体制的密钥交换协议(见 13.3.3.4 节)和 SOK 密钥分享系统(见 13.3.5 节)为什么能够抗击中间人攻击?
- 13.10 由式(13.3.4)我们知道修正的 Weil 对是对称的。对于线性独立的两个点,最初的 Weil 对是否也是对称的?

提示:使用双线性性和同一性  $e_a(P + X, P + X)$ 。

- 13.11 如果我们把 SOK 密钥分享系统(见 13.3.5 节)看做是 Diffie-Hellman 密钥交换协议的一个非交互形式,那么 Boneh 和 Franklin 的基于 ID 的密码系统(算法 13.2)是否可以看做是交互密码系统的非交互形式?

提示:通常的公钥密码系统是交互的,也就是说,在加密和发送密文之前,发送者必须首先得到接收者的公钥。





# 第五部分 建立安全性的 形式化方法

系统的复杂性是计算与通信系统失败的主要原因,尤其是由系统与其组成部分之间的通信所引发的复杂性。密码学系统(算法和协议)通常都是复杂的系统,而且它们的失败有一个更重要的原因:攻击。普通计算与通信系统的工作环境是友好的(为了避免系统失败,用户都尽可能仔细地不给系统中的程序或通信伙伴输入和发送无效的数据),而密码学系统的工作环境却充满了敌意。除了普通方式的可能失败以外,密码学系统还会因为精心策划的异常使用而失败。它们遭受着各种各样的攻击,这些攻击可能是没有被邀请但可以与系统交互信息的系统外部攻击者发起的,也可能是系统内部的合法用户发起的。即使是由专家设计的密码学系统也经常失败。我们不能过于相信安全专家,Needham-Schroeder 协议中长期隐藏的缺陷就给我们上了很好的一课(见第 2 章)。

系统分析的形式化方法包括一些分析任务的系统过程。为了在模型化和建立复杂系统的过程中,并且在观察和推理复杂系统的行为时,保持严格性和一般性,这些系统过程都直接构建在数学的基础上。它们或者以一种有系统的方式设计一个系统,使希望获得的系统特性易于证明,或者通过一个系统化的查找方法检查系统,找到系统的错误。密码体制易于失败的本质使人们广泛地认为这些系统应该通过形式化的方法进行设计和/或分析。

本书这一部分以设计和分析密码学系统的形式化方法为主题,分为四章。第 14 章介绍公钥密码体制强(即适于应用)安全性概念的形式化定义。我们将从教科书式安全性概念层层递进到适于应用的安全性概念。第 15 章介绍和说明两个重要而且实用的公钥密码体制,它们的适于应用安全性是使用第 14 章中定义的概念为基础建立起来的<sup>①</sup>。第 16 章介绍数字签名的一个适于应用的安全性概念,在这个强安全性概念下,我们将描述对几个签名方案证明安全性的一些技巧。第 17 章我们将再次讨论认证协议这个主题,介绍各种形式化分析认证协议正确性的技巧。

---

<sup>①</sup> 原文可能有误,漏掉了“Chapter 14.”

## 第 14 章 公钥密码体制的形式化强安全性定义

### 14.1 引言

保密性是密码学的核心。本章我们将研究公钥加密算法的安全性概念。在后面的章节中,我们将看到本章建立的关于保密的安全性概念更具一般性,它是建立其他各种安全性服务的基础。

到目前为止,我们还只限于讨论非常弱的公钥密码体制的保密性概念,这个概念在性质 8.2(见 8.2 节)中描述:我们只考虑了被动攻击者,这种攻击者只是在“完全或无”的意义下攻破目标密码体制。它是一个典型的教科书式密码学安全性概念,不适合实际应用。

实际中的攻击者更可能是主动的,同时还拥有被动窃听的能力,即他们可以用某种不确定的方式修改密文或者计算明文,将所得结果发送给一个无心的用户以得到预言机服务(见 7.8.2.1 节和 8.9 节)。因此,只考虑被动攻击者的安全性概念是不够强的。我们需要预先考虑 Malice,即一个主动而且聪明的攻击者(关于 Malice 的能力请回顾 2.3 节)。

另外,在很多应用中,明文消息可能包含一些容易猜测的先验信息。例如,这些先验信息可能是已知薪水范围中的一个值、投票协议中几个已知候选人中的一个、几条可能指示中的一条,或者甚至是有关明文的 1 比特的信息(如给股票经纪人的购买/抛售指令,远程“掷币”协议 1.1 中的正面/反面)。对于使用一个基于陷门单向函数的公钥加密算法加密的明文,为了猜测与之相关的这类信息,Malice 可以简单地加密他所猜测的明文,比较加密结果是否与目标密文一样即可。因此,在“完全或无”意义下的保密性概念是不够强的。

为了预先考虑不同程度的攻击,我们需要各种更严格的安全性概念。而为了建立更严格的安全性概念,我们需要做的第一步就是要正确地形式化需要解决的问题。在具有形式化可证明安全性的密码学系统范围内,人们提出了各种各样的攻击游戏来模型化并获得各种攻击情形。这些游戏是在 Malice 和随机预言机之间进行的。游戏的规则允许 Malice 获得由随机预言机提供的密码学帮助,Malice 也确实需要这些帮助。我们可以将这些帮助看做是为 Malice 提供了一种“密码分析训练课程”。对于一个攻击游戏的形式化模型,如果即使给 Malice 足够的“密码分析训练课程”,他也不能获得满意的成功,就认为这个密码体制是安全的。

形式化处理安全性的另一个方面是对 Malice 满意程度的严格度量。在密码学系统的可形式化证明安全性领域,密码系统的安全性涉及一个量化的关系,这个关系把该密码系统的安全性与计算复杂度理论中的某一个难题联系起来。对安全性建立高度信任的标准方法是把 Malice 对攻破目标密码体制的满意程度表达和转换成某些数值,这些数值将度量人们解决计算复杂度理论中某些著名难题需要多少时间、成功的概率有多大。这样的转换实际上是一个有效的数学变换,或者一系列这一类变换,将所谓的成功攻击转换成一某个著名难题的解。因为我们非常相信解决该难题需要很长时间,成功的也概率很小,所以我们也十分确信 Malice 对攻破目标密码体制的所谓攻击不满意。

由于攻击情形各种各样,而且要用几个困难问题作为安全性的计算基础,所以具有形式化可证明安全性的密码学系统领域内采用了一种有很多专用术语的语言。我们可以用这些专用术语方便而精确地描述各种不同的安全特性和要求。这里我们将列举几个关于安全性陈述的例子:

- 密码体制  $X$  对于拥有无限计算资源的被动窃听者是语义安全的,但在选择密文攻击下是可展的,此时攻击者只须在午餐时间用一个廉价设备工作,如掌上电脑。
- 数字签名方案  $Y$  在适应性选择消息攻击下关于签名的不可伪造性是安全的。该安全性的形式化证明将一个成功的伪造归约为对离散对数问题的解。这个归约借助于随机预言机模型,在该模型中伪造者被分叉的概率是不可忽略的。
- 签密方案  $Z$  在适应性选择密文攻击下关于加密的不可区分性以及在选择消息攻击下关于签名的不可伪造性是安全的,无论攻击者是在午餐时间、午夜,还是在凌晨。这些安全特性的形式化证明都是对于整数分解问题而言的。
- 电子拍卖协议  $\Pi$  对投标者在不可否认他们参与协议运行的意义下,以及对中标者在其身份不可区分的意义下,是安全的。这些安全特性的形式化证明是对于一个标准的困难问题假设(判定 Diffie-Hellman 问题)而言的。

在我们对本章和后续三章关于形式化可证明安全性的研究过程中,对以上安全性陈述中出现的专用术语将给出详细定义和说明。在我们的研究之后,像上面所列举的安全性陈述就会变得更有意义。

### 14.1.1 本章概述

在 14.2 节我们开始介绍本章的主题:对安全性的形式化处理,这包括对攻击情景的形式模型化和对所得结论的精确度量。这种形式化处理明确表明第 8 章介绍的基于“完全或无”的安全性概念是不充分的。一个强化的安全性概念“语义安全性”将在 14.3 节中介绍,它意味着隐藏有关消息的任何部分信息。语义安全性的不充分性将在 14.4 节中给出。这就使我们得到了进一步强化的安全性概念:“选择密文的安全性”、“适应性选择密文的安全性”和“不可延展性”。这些强化的安全性概念及它们之间的关系将在 14.5 节中讨论。

## 14.2 安全性的形式化处理

抽象地讲,对于密码学系统,“形式化可证明安全性”是对一个系统抗击攻击能力的一种肯定的量度。安全的系统是那些 Malice 不能过于频繁或足够快地做坏事的系统。所以,度量的内容包括成功概率和计算代价。

为了对这一度量有一个具体认识,让我们看一个在 Malice 和“预言机”之间进行的“攻击游戏”,该预言机模仿密码系统中的一个天真用户,其中用户不可避免地要与 Malice 进行交互。这个攻击游戏为我们提供了对计算观点的所谓安全性的一种形式化论述;同时它也是我们强化密码体制安全性概念这一过程(从 8.2 节性质 8.2 中的概念开始)中的第一步。

假设 Malice 是攻击者, $\mathcal{O}$  表示攻击游戏中的预言机。在保密环境中, Malice 的目标是一个密码体制。于是,“坏事情”就意味着该游戏会破坏目标密码体制的保密性。

我们用 7.2 节中的定义 7.1 来描述目标密码体制,这包括一个加密算法 $\mathcal{E}$ 、明文空间 $\mathcal{M}$ 和密文空间 $\mathcal{C}$ 。但是这里我们要注意一点:现在加密算法 $\mathcal{E}$ 是概率的,也就是说,它有一个服从某种概率分布的内部随机操作,使输出的密文作为一个随机变量也服从该分布。例如,如果一条明文消息在同一个加密密钥下加密两次,那么(由加密算法的一一映射特性)将以“压倒性”概率得到两条不同密文。

协议 14.1 详细描述了一个攻击游戏。

#### 协议 14.1 不可区分的选择明文攻击

假定

- i) Malice 和一个预言机 $\mathcal{O}$ 商定了一个目标密码体制 $\mathcal{E}$ ,它的明文消息空间是 $\mathcal{M}$ ,密文消息空间是 $\mathcal{C}$ ;
- ii)  $\mathcal{O}$ 固定了 $\mathcal{E}$ 的一个加密密钥  $ke$ 。

1. Malice 选择两条不同的消息  $m_0, m_1 \in \mathcal{M}$ ,将它们发送给 $\mathcal{O}$ ;

(\* 消息  $m_0, m_1$  称为选择明文消息。Malice 到现在为止还处于准备  $m_0, m_1$  的“寻找阶段”:她当然希望准备加密后易于识别的两条明文\*)

2. 如果这两条消息不同长, $\mathcal{O}$ 就把短的消息扩充,使其与另一个一样长;

(\* 例如,扩充  $d$  个  $0^d$  是两条消息长度之差\*)

$\mathcal{O}$ 投掷一个公平硬币  $b \in_U \{0,1\}$ ,然后执行下列加密操作

$$c^* = \begin{cases} \mathcal{E}_{ke}(m_0) & \text{若 } b = 0 \\ \mathcal{E}_{ke}(m_1) & \text{若 } b = 1 \end{cases}$$

$\mathcal{O}$ 向 Malice 发送  $c^* \in \mathcal{C}$ ;

(\* 密文  $c^*$  称为询问密文。和可证明安全性中的习惯一样,总是把带上标“\*”的密文看做是询问密文\*)

(\* 记住, $c^*$  是关于两个随机输入值的随机变量:公平硬币  $b$  和 $\mathcal{E}$ 的内部随机操作\*)

3. 收到  $c^*$  后, Malice 必须回答 0 或 1,作为他对 $\mathcal{O}$ 的硬币投掷结果的猜测。

(\* Malice 现在处于对 $\mathcal{O}$ 的硬币投掷的有根据“猜测阶段”;只允许回答 0 或 1\*)

在攻击游戏中, $\mathcal{O}$ 询问 Malice,要求其回答下述问题:

询问密文  $c^*$  来自总体(实验) $\mathcal{E}_{ke}(m_0)$ 还是 $\mathcal{E}_{ke}(m_1)$ ?

将 Malice 看做是定义 4.14(见 4.7 节)中的概率多项式时间区分器,这是因为 $\mathcal{O}$ 的输出是概率的,而且 Malice 是多项式时间有界的,当他认为一个概率多项式时间(PPT)的算法可能比一个确定的算法更有效时(我们在第 4 章曾多次看到通常都是如此),他就会用 PPT 算法。用 Adv 表示 Malice 区分 $\mathcal{E}_{ke}(m_0)$ 和 $\mathcal{E}_{ke}(m_1)$ 的概率优势。由定义 4.14, Adv 应该是 Malice 概率区分总体 $\mathcal{E}_{ke}(m_0)$ 和 $\mathcal{E}_{ke}(m_1)$ 的差:

$$\text{Adv} = |\text{Prob}[0 \leftarrow \text{Malice}(c^* = \mathcal{E}_{ke}(m_0))] - \text{Prob}[0 \leftarrow \text{Malice}(c^* = \mathcal{E}_{ke}(m_1))]|. \quad (14.2.1)$$

概率空间应该包括  $\mathcal{O}$ 、Malice 和加密算法的内部随机操作所做的概率选择。还要注意 Malice 的回答不仅仅依赖于询问密文  $c^*$ , 还依赖于所选择的两条明文消息  $(m_0, m_1)$ 。正因为如此, 我们可以把他的回答看做是一个“有根据的猜测”。但是, 为了简化说明, 我们在 Malice 的输入中省略了  $(m_0, m_1)$ 。

必须注意 Malice 还有一个线索“改进”其有根据的猜测:  $\mathcal{O}$  投掷的是一个公平硬币。虽然我们无疑知道式(14.2.1)中的每一个概率项都不能超过  $1/2$ , 例如, 事件“ $c^* = \mathcal{E}_{ke}(m_0)$ ”发生的概率只有  $1/2$ , 但是优势公式(14.2.1)并没有明确地表明 Malice 如何使用这一线索。我们应该在 Malice 的优势公式中表明这一点。注意到  $\mathcal{O}$  的硬币投掷在两种情况的概率是相等的, 都是  $1/2$ , 应用条件概率(3.4.1 节中的定义 3.3), 式(14.2.1)可以写为

$$\text{Adv} = \left| \frac{1}{2} \text{Prob}[\mathcal{O} \leftarrow \text{Malice}(c^*) \mid c^* = \mathcal{E}_{ke}(m_0)] - \frac{1}{2} \text{Prob}[\mathcal{O} \leftarrow \text{Malice}(c^*) \mid c^* = \mathcal{E}_{ke}(m_1)] \right| \quad (14.2.2)$$

由游戏的规则, 除了 0 和 1, 不允许 Malice 有其他回答, 因此回答错误这一事件是回答正确的补。这样, 由概率的性质 5(见 3.3 节), 有

$$\text{Adv} = \left| \frac{1}{2} \text{Prob}[\mathcal{O} \leftarrow \text{Malice}(c^*) \mid c^* = \mathcal{E}_{ke}(m_0)] - \frac{1}{2} (1 - \text{Prob}[\mathcal{O} \leftarrow \text{Malice}(c^*) \mid c^* = \mathcal{E}_{ke}(m_0)]) \right|$$

即

$$\text{Prob}[\mathcal{O} \leftarrow \text{Malice}(c^*) \mid c^* = \mathcal{E}_{ke}(m_0)] = \frac{1}{2} \pm \text{Adv} \quad (14.2.3)$$

式(14.2.3)经常用于表示一个算法在纯粹的公平硬币投掷猜测(概率  $1/2$ )之上的优势。所以, 若  $\text{Adv} = 0$ , 则 Malice 的概率回答就正好与投掷公平硬币具有相同的分布。当然, 我们不应该过于挑剔 Malice 算法的优势, 而应考虑(i)  $\text{Adv} > 0$  和(ii)  $\text{Adv}$  前的加号。显然, 式(14.2.3)中的 0 换成 1 也成立。

从式(14.2.3)我们还可以看出 Malice 的优势不能超过  $1/2$ , 这是因为概率值不能超出区间  $[0, 1]$ 。事实上, 假设预言机  $\mathcal{O}$  正好以  $1/2$  的概率加密两个明文中的一个, 那么式(14.2.1)中的  $\text{Adv}$  作为联合事件的概率差也不能超过  $1/2$ 。读者可能希望知道如果  $\mathcal{O}$  投掷一个有偏硬币, 例如以  $1/4$  的概率加密提问明文  $m_0$ ,  $3/4$  的概率加密  $m_1$ , 式(14.2.3)会是怎样的? 提示: 将式(14.2.2)中的  $1/2$  分别替换为有偏的概率值, 看式(14.2.3)如何变化。此时, 我们将会看到只要  $\mathcal{O}$  投掷一个有偏硬币, Malice 的优势就有可能超过  $1/2$ 。

如果  $\mathcal{E}_{ke}(m_0)$  和  $\mathcal{E}_{ke}(m_1)$  是不可区分的, 我们就称目标密码体制对于协议 14.1 中的攻击游戏是安全的。根据定义 4.15(见 4.7 节), 这就意味着不存在优势大于 0 且不可忽略的 PPT 区分器。等价地, 对于任意能成功区分的 Malice, 他的优势都是一个可忽略的量。这里的“可忽略”由目标加密方案的安全参数度量, 该参数通常是密钥材料的大小。我们可以把任意多项式有界(即任何 PPT 算法)的 Malice 的优势  $\text{Adv}$  看做是关于他的计算资源的慢增函数。这里“慢增”的含义是: 即使 Malice 惊人地增加他的计算资源, 也只能使优势取得很小的增长, 以至于 Malice 不满意自己的“优势”。这也正是在本章开始部分提到的 Malice 不能过于频繁或足够快地做坏事的真正含义。



因为我们的讨论正好遵循多项式不可区分总体的定义 4.15, 所以我们刚才定义的新安全性概念就称为加密的多项式不可区分性。另外, 由于 Malice 选择的两条明文之间的不可区分性, 所以这个新概念的精确名称应该是多项式不可区分选择明文攻击的安全性, 通常简写为 IND-CPA 安全性。

在协议 14.1 的 IND-CPA 攻击游戏中, Malice 可以自由选择明文消息, 而且只要回答有关选择明文的 1 比特信息: “加密的消息是  $m_0$  还是  $m_1$ ”, 那么 Malice 攻破目标密码体制的难度就显著地降低, 从该 IND 问题的难度降为在“完全或无”安全意义(定义于 8.2 节的性质 8.2(i)) 下攻破该体制。事实上, 到现在为止, 我们已经介绍过的所有教科书式公钥加密算法(关于教科书密码学的含义参考 8.14 节)在 IND-CPA 下都不安全。对 RSA 和 Rabin 密码体制, 我们很容易看到这一点, 因为它们都是确定性的, 从而 Malice 可以通过再加密准确指出是  $m_0$  还是  $m_1$ 。在 14.3.5 节中, 我们将进一步看到算法 8.3 中给出的 ElGamal 体制虽然提供了一个概率加密算法, 但在 IND-CPA 下也是不安全的。

随着攻击难度的降低, 我们需要加强对密码体制的安全要求。本章的主旨就是用严格的形式化方法降低 Malice 攻击密码体制的难度, 或等价地说, 加强密码体制的安全性概念。

### 14.3 语义安全性——可证明安全性的首次亮相

前面定义的 IND-CPA 安全性概念最初是由 Goldwasser 和 Micali 在[127]中介绍的。他们命名为语义安全性。语义安全性意味着密文不会向任何计算能力为多项式有界的敌手泄漏有关相应明文的任何有用信息(如果我们认为明文的长度不是有用信息)。他们发现, 在很多的应用中消息可能含有某些对于攻击非常有利的先验信息。例如, 密文也许只是对一个简单的“买”或“卖”指示的加密, 或者是对参与选举的已知候选人的身份之一加密。Goldwasser 和 Micali 指出直接应用陷门单向函数的公钥密码体制一般都不能隐藏这样的消息。我们将看到他们的批评适用于第 8 章中介绍的所有公钥密码体制。

对这种更强安全性概念的需要是非常现实的。智力扑克协议的失败就很好地说明了直接应用陷门单向函数的公钥密码体制的缺陷。我们首先回顾一下 Shamir、Rivest 和 Adleman 的智力扑克协议。

#### 14.3.1 SRA 智力扑克协议

Alice 住在纽约, Bob 住在伦敦。他们尚未谋面, 但他们希望跨越大西洋玩扑克。又是 RSA 体制的创始人 Shamir、Rivest 和 Adleman 使这成为可能, 他们提出了“SRA 智力扑克”协议。

智力扑克与普通扑克的玩法相同, 但是为了可以在通信中玩这个游戏, 要把扑克牌加密成消息。为了玩扑克游戏, Alice 和 Bob 首先应该公平发牌。这里的“公平”包括以下四点要求:

- i) 发牌必须等可能地分配所有可能的各手牌(即均匀分布), 而且不允许同一张牌同时出现在两个人手中。
- ii) Alice 和 Bob 必须知道自己手中的牌, 但他们都不能知道有关对方手中牌的任何信息。
- iii) 必须把 Alice 和 Bob 看做是潜在的欺骗者, 我们不能指望他们会遵守协议的规则。
- iv) Alice 和 Bob 都可以验证正在进行的游戏是公平的。

SRA 智力扑克的思想是利用具有交换特性的密码。在这样的密码中,消息可以由 Alice 和 Bob 分别用他们的秘密密钥双重加密,得到的密文也必须由他们俩双重解密。令

$$C = E_X(M) \text{ 和 } M = D_X(C)$$

表示由主体  $X$  根据规则执行的加解密算法。该密码的交换特性是对于明文空间中的任意  $M$ , 以下两个等式都成立:

$$\begin{aligned} M &= D_A(D_B(E_A(E_B((M)))))) \\ &= D_B(D_A(E_B(E_A((M)))))) \end{aligned} \quad (14.3.1)$$

也就是说,即使双重解密序列与双重加密序列相互独立,也能正确恢复明文消息。

为了简单又不失一般性,我们假设 Alice 和 Bob 决定使用一副只有三张牌的扑克玩人手一张的游戏。协议 14.2 详细说明了一种公平发牌的方法。对于一副牌有很多张的一般情况也可以直接给出,但是很繁琐。

#### 协议 14.2 SRA 智力扑克游戏的一个公平发牌协议

假定:

Alice 和 Bob 商定了一个具有式(14.3.1)中性质的交换密码,并且每人选定了各自的秘密加密密钥;

他们还商定了一副有三张牌  $M_1, M_2, M_3$  的扑克。

目标:

他们得到人手一张的公平发牌,满足公平性质(i)~(iv)。

1. Alice 将三张牌加密为  $C_i = E_A(M_i), i = 1, 2, 3$ ; 以随机的顺序将它们发给 Bob;  
(\* 以随机的顺序将牌发送给 Bob 等于洗牌 \*)
2. Bob 随机选一条密文,记为  $C$ , 将  $C$  加密为  $CC = E_B(C)$ ; 再随机选另一密文,记为  $C'$ ; 将  $CC$  和  $C'$  发送给 Alice;  
(\*  $CC$  确定了 Bob 手中的牌;  $C'$  确定了 Alice 手中的牌; 另一张牌丢弃 \*)
3. Alice 解密  $CC$  和  $C'$ ;  $C'$  解密后是她手中的牌;  $CC$  的解密,记为  $C''$ , 返回给 Bob;
4. Bob 解密  $C''$ , 从而得到他手中的牌。  
(\* 现在他们可以玩智力扑克游戏了 \*)

#### 14.3.2 基于教科书式安全的安全性分析

从现在开始我们假设不管是单加密还是双加密,协议 14.2 中使用的密码体制都是足够强的。我们称一个密码体制是“足够强的”,意味着给定一条明文(密文),如果加密密钥(解密密钥)未知,那么多项式有界的攻击者就不能从给定明文构造出一个有效的密文(不能从给定密文恢复出相应的明文)。这是我们对教科书密码算法(见 8.2 节)已经认可的、在性质 8.2(i)中给出的“完全或无”意义下的秘密性。在这一安全性概念下,我们可以针对公平性质(i)~(iv)给出协议 14.2 的安全性分析。

协议 14.2 运行一次后:

- Alice 和 Bob 以相等的概率各自获得  $\{M_1, M_2, M_3\}$  中的一张牌(即在该集合中均匀分布):这是因为 Alice 在步骤 1 中洗了牌。注意 Alice 所关心的是均匀随机洗牌,以防止 Bob 选择他手中牌的优势。所以公平性质(i)成立。
- 在双解密之后,双方都知道自己手中的牌,但由于他们都不知道丢弃的牌是哪一张,所以他们不知道对方手中的牌。这样,性质(ii)成立。
- 显然该协议不依赖于任何一方的诚实性,所以性质(iii)成立。

公平性质(iv)取决于协议中使用的密码体制是否允许游戏后的诚实验证。Shamir 等人建议使用 RSA 体制(见 8.5 节)的一个变形,在游戏结束前,双方不仅保密他们的加密指数而且保密他们的解密指数,当游戏结束后,向对方揭示这些指数以验证他们的行为是否诚实。

假设  $N$  是共享的 RSA 模数。在变形中, Alice 和 Bob 都知道  $N$  的分解。令  $(e_A, d_A)$  是 Alice 的加解密指数,  $(e_B, d_B)$  是 Bob 的加解密指数。因为 Alice(Bob)知道  $N$  的分解式,所以她可以从  $e_A$  计算出  $d_A$  (他从  $e_B$  计算出  $d_B$ )。这可以通过解同余式

$$e_X d_X \equiv 1 \pmod{\phi(N)} \quad (14.3.2)$$

得到( $X$  是  $A$  或  $B$ )。对参与者  $X$ , 有

$$E_X(M) = M^{e_X} \pmod{N}$$

$$D_X(C) = C^{d_X} \pmod{N}$$

因为 RSA 群是交换群,所以不难看出式(14.3.1)成立。在游戏结束前,双方都保密他们的加解密指数,所以任何一方都不能生成由另一方生成的有效密文,这可以阻止任何一方测试哪一个密文是哪一张牌的加密。而且,任何一方也不能解密由另一方生成的密文。所以,正如我们要求的,该密码体制确实是“足够强”的。

现在很清楚,游戏结束后,双方都可以向对方揭示他们的加解密指数,因此他们可以检查加密、双加密和解密是否都正确执行。所以,公平性质(iv)成立。

在我们的分析中,我们使用了一个不充分也不合理的安全性概念:“足够强”。一个“足够强”的密码体制意味着攻击者不能在不知道正确加密密钥的情况下,对给定明文生成一个有效密文,或者在不知道正确解密密钥的情况下,解密一个密文。但是现在“足够强”这一安全性概念的不充分性和不合理性就变得很明显了。Lipton[180]发现如果协议 14.2 使用由智力扑克游戏最初创始者建议的 RSA 体制的变形,它就会失败。失败的原因是该体制不能隐藏明文消息中的某种先验信息:二次剩余。回顾 6.5 节,一个数  $a$  是模  $N$  的二次剩余,如果  $\gcd(a, N) = 1$ , 且存在  $x < N$ , 使得

$$x^2 \equiv a \pmod{N}$$

注意因为  $\phi(N)$  是偶数,所以满足式(14.3.2)的加解密指数  $e$  和  $d$  必须都是奇数。因此,明文  $M$  是模  $N$  的二次剩余,即  $M \in QR_N$ , 当且仅当相应的密文  $C \in QR_N$ , 因为对于某个  $x < N$ , 有

$$C \equiv M^e \equiv (x^2)^e \equiv (x^e)^2 \pmod{N}$$

也就是说, RSA 加密不能改变明文的二次剩余特性。再回顾 6.5 节,我们知道由  $N$  的分解式判定  $C$  是否属于  $QR_N$  是很容易的:首先对  $C$  约简模  $N$  的每一个素因子,然后用算法 6.2 计算这些结果的 Legendre 符号。

因此,如果某一张(两张)明文牌在  $QR_N$  中,另两张(一张)不在  $QR_N$  中,那么知道 Lipton 技巧的一方就会在游戏中占有不公平的优势:他(她)确切地知道对于哪一张牌加密等于没有加密,不管是单加密还是双加密。

我们的结论是 SRA 智力扑克协议是不安全的。对这一结论的精确形式化陈述是,该协议对于协议 14.1 中描述的 IND-CPA 模型是不安全的。

### 14.3.3 Goldwasser 和 Micali 的概率加密

为了抵抗 Lipton 攻击,我们可以修改协议 14.2。例如,要求所有的牌都从  $QR_N$  中选择就是一种具体的修改方法。但是,Goldwasser 和 Micali 预想到需要针对一个更大的问题进行普遍的修改,这个问题就是需要更强的安全性概念——语义安全性,他们对该概念的描述如性质 14.1。

**性质 14.1 语义安全性** 凡是在给定密文条件下可以有效计算的有关相应明文的信息,都可以在没有该密文的条件下有效地计算。

他们提出了一个具有该性质的概率加密方案,我们把它命名为 GM 密码体制。GM 体制对整个消息逐比特加密,其中从密文找出一比特加密消息的难度等于确定  $c \in QR_N$  或  $c \in J_N(1) \setminus QR_N$  的难度,这里  $J_N(1) = \{x \mid x \in \mathbb{Z}_N^*, \left(\frac{x}{N}\right) = 1\}$ 。

GM 体制在算法 14.1 中详细介绍。

#### 算法 14.1 Goldwasser 和 Micali 的概率密码体制

##### 密钥建立

为了建立用户的密钥材料,用户 Alice 执行下列步骤:

1) 选择两个随机素数  $p$  和  $q$ , 满足  $|p| = |q| = k$ ;

( \* 如使用输入为  $1^k$  的算法 4.7 \* )

2) 计算  $N = pq$ ;

3) 选择一个随机整数  $y$ , 满足  $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$ ;

( \* 使得  $y \in J(N) \setminus QR_N$  \* )

4) 公开  $(N, y)$  作为她的公开密钥, 其私钥为  $(p, q)$ 。

##### 加密

为了给 Alice 发送二进串  $m = b_1 b_2 \cdots b_\ell$ , Bob 执行:

for( $i = 1, 2, \cdots, \ell$ )

{

$x \leftarrow_U \mathbb{Z}_N^*$ ;

if( $b_i = 0$ )  $c_i \leftarrow x^2 \pmod{N}$

else  $c_i \leftarrow yx^2 \pmod{N}$

}

Bob 将  $E_N(m) \leftarrow (c_1, c_2, \cdots, c_\ell)$  发送给 Alice。

**解密**

收到密文  $(c_1, c_2, \dots, c_\ell)$  后, Alice 执行:

```

for(  $i = 1, 2, \dots, \ell$  )
{
    if(  $c_i \in QR_N$  )  $b_i \leftarrow 0$ 
    else  $b_i \leftarrow 1$ ;
}
set  $m \leftarrow (b_1, b_2, \dots, b_\ell)$ .

```

现在我们证明算法 14.1 中介绍的体制确实是一个密码体制, 即 Alice 解密过程返回的明文确实与 Bob 加密的是同一个。

观察该加密算法, 易于看出明文比特 0 被加密成  $QR_N$  中的一个元素。

对于明文比特 1, 相应的密文为  $c = yx^2$ 。注意  $\left(\frac{y}{p}\right) = \left(\frac{y}{q}\right) = -1$ , 所以(由 Legendre 符号的乘法交换性质, 6.5.2 节中的定理 6.16)有:

$$\left(\frac{c}{p}\right) = \left(\frac{yx^2}{p}\right) = \left(\frac{y}{p}\right) \left(\frac{x^2}{p}\right) = (-1) \times 1 = -1$$

和

$$\left(\frac{c}{q}\right) = \left(\frac{yx^2}{q}\right) = \left(\frac{y}{q}\right) \left(\frac{x^2}{q}\right) = (-1) \times 1 = -1$$

因此

$$\left(\frac{c}{N}\right) = \left(\frac{c}{p}\right) \left(\frac{c}{q}\right) = (-1) \times (-1) = 1$$

也就是说明文比特 1 被加密为  $J_N(1) \setminus QR_N$  中的一个元素。

解密算法工作正确, 因为 Alice 知道  $p, q$ , 所以她可以确定  $c_i \in QR_N$  还是  $c_i \in J_N(1) \setminus QR_N$ , 从而可以正确地逐比特恢复明文。

不难看出加密  $\ell$  比特消息  $m$  需要  $O_B(\ell(\log_2 N)^2)$  比特操作, 这是加密的时间复杂度。加密算法的消息扩展率为  $\log_2 N$ , 即一比特明文被加密成  $\log_2 N$  的密文。

因为计算模  $p$  和模  $q$  ( $|p| = |q| = k$ ) 的 Legendre 符号需要  $O_B(k^2)$  比特操作(回顾算法 6.2 之后对于计算 Jacobi 符号算法详细实现的讨论),  $(c_1, c_2, \dots, c_\ell)$  的解密需要  $O_B(\ell(\log_2 N)^2)$  比特操作, 这就是解密的时间复杂度。

逐比特加密的方式表明 GM 体制的效率很低。

#### 14.3.4 GM 密码体制的安全性

GM 体制的加密算法可以看做是一个无差错的随机化算法: 加密算法中的随机操作不会在密文中引入任何错误, 但却获得了下面的重要功能:

将明文比特 0 均匀分布在  $QR_N$  上, 明文比特 1 均匀分布在  $J_N(1) \setminus QR_N$  上。

两个分布都是均匀分布,因为对于明文比特 0,二次方运算是  $\mathbb{Z}_N^*$  到  $QR_N$  的满射,而对于明文比特 1,对  $QR_N$  中的元素乘以  $y$  是从  $QR_N$  到  $J_N(1) \setminus QR_N$  上的双射。所以,加密算法中选择  $x \in \mathbb{Z}_N^*$  意味着或者当明文比特是 0 时,在  $QR_N$  中均匀地选择一个元素,或者当明文比特是 1 时,在  $J_N(1) \setminus QR_N$  中均匀地选择一个元素。

正式地讲,GM 体制的难度是判定二次剩余特性的问题(QR),该问题在定义 6.2(见 6.5.1 节)中详细描述过。QR 问题是数论中的一个著名难问题(回顾我们在 6.5.1 节中定义 6.2 之后的讨论)。关于它的难度,我们有以下假设。

**假设 14.1 二次剩余假设(QR 假设)** 令  $IG$  是一个整数实例生成器,输入为  $1^k$ ,运行时间是关于  $k$  的多项式,输出一个  $2k$  比特的模数  $N = pq$ ,  $p$  和  $q$  是均匀分布的随机素数。

我们称  $IG$  满足二次剩余(QR)假设,如果对所有足够大的  $k$  以及  $N \leftarrow IG(1^k)$ ,总体  $QR_N$  和  $J_N(1) \setminus QR_N$  是多项式不可区分的(见 4.7 节定义 4.14)。

显然获得公钥  $N$  为 QR 问题的难度提供了一个上界,因为攻击者只要分解  $N$ ,然后应用 GM 解密算法解 QR 问题即可。因此,GM 体制假设攻击者是多项式有界的。这也是加密算法的语义安全性也被称为加密的多项式不可区分性的原因。

如果 QR 假设确实成立,那么我们可以认为从一个多项式有界的攻击者角度看,GM 加密算法将一个明文比特均匀地分布到密文空间  $J_N(1)$  中。密文的均匀分布意味着这类攻击者从密文猜测相应明文的企图是完全没有意义的。这正是 Goldwasser 和 Micali 以性质 14.1 的形式给出的语义安全性概念的含义。

**定义 14.1 语义安全性,不可区分性选择明文攻击的安全性(IND-CPA 安全性)** 称一个安全参数为  $k$  的密码体制是语义安全的(IND-CPA 安全的),如果任何多项式有界的攻击者进行协议 14.1 中的攻击游戏之后,式(14.2.3)给出的  $\text{Adv}$  都是关于  $k$  的一个可忽略量。

对于 GM 体制的安全性,我们有下面的结论。

**定理 14.1** 假设  $k$  为 RSA 模数  $N$  的两个素因子的规模,则参数为  $k$  的 GM 密码体制是语义安全的(IND-CPA 安全的)当且仅当 QR 假设成立。  $\square$

### 14.3.5 ElGamal 体制的一种语义安全版本

与 RSA 的情况类似,算法 8.3 中介绍的 ElGamal 体制也不能隐藏明文的二次剩余特性。这是因为在该算法中,我们设定了公开参数  $(g, p)$ ,  $g$  是整个群  $\mathbb{Z}_p^*$  的生成元。在这种参数背景下,明文的二次剩余特性可以与对应的密文联系起来。例 14.1 将给出了这一联系。

**例 14.1** 假设随机预言机  $\mathcal{O}$  为算法 8.3 中的 ElGamal 体制建立了  $(p, g, y)$  作为公钥材料,则由 Euler 准则(见 6.5.1 节中的定理 6.13),  $g \in QNR_p$  (即  $g$  是一个模  $p$  的非二次剩余)。

假设 Malice 是一个 IND-CPA 攻击者。他可以提交一条消息  $m_0 \in QR_p$ , 一条消息  $m_1 \in QNR_p$  (应用算法 6.2, Malice 很容易准备出满足这两个条件的  $m_0$  和  $m_1$ )。设  $(c_1^*, c_2^*)$  是从  $\mathcal{O}$  返回的询问密文对,则有

$$c_2^* = \begin{cases} y^k m_0 \pmod{p} & 50\% \text{ 的概率} \\ y^k m_1 \pmod{p} & 50\% \text{ 的概率} \end{cases}$$



现在 Malice 可以通过判定  $y, c_1^*$  和  $c_2^*$  的二次剩余特性, 明确指出被加密的明文。下面是需要考虑的几种情况。

首先我们考虑  $y \in \text{QR}_p$  的情况, 非常简单。被加密明文是  $m_0$  当且仅当  $c_2^* \in \text{QR}_p$ 。这可以由定理 6.16(ii)(见 6.5.2 节)给出的 Legendre 符号的乘法交换特性得到。

对于  $y \in \text{QNR}_p$ , 需要考虑两种简单的情况。第一种情况是  $c_1^* \in \text{QR}_p$ , 这将导致  $y^k \in \text{QR}_p$  (因为现在  $k$  是偶数), 因此判定准则与前一段中的相同。读者可以自己完成第二种情况,  $c_2^* \in \text{QNR}_p$ , 注意此时  $k$  是奇数。□

和通常一样, 只要我们发现了问题之所在, 修正就很容易。如果我们限制该密码体制只在  $\text{QR}_p$  中工作, 那么例 14.1 中的攻击就不会成功了。算法 14.2 具体给出了一种修改。

首先我们应该注意到算法 14.2 最后将终止, 因为有很多素数  $p$  满足  $(p-1)/2$  也是素数 (例如 7, 11, 23, 39, 47)。这样的素数称为安全素数。

其次, 由费马小定理,  $\text{ord}_p(g) = q$  是一个大素数, 因此, 群  $G = \langle g \rangle$  的阶数很大。这是 DL 假设(假设 8.2)成立的一个必要条件。

#### 算法 14.2 ElGamal 体制的一种语义安全变形

##### 公开参数建立

假设  $G$  是一个阿贝尔群, 其描述如下:

1. 找一个随机素数  $q, |q| = k$ ;
2. 检测  $p = 2q + 1$  的素性, 如果  $p$  不是素数, 返回 1;
3. 选择一个随机生成元  $h \in \mathbb{Z}_p^*$ ; 令  $g = h^2 \pmod{p}$ ;
4. 令  $\text{desc}(G)$  是群  $G = \langle g \rangle$  的描述;  
( \* 由  $g$  生成的群, 见 5.2.3 节的定义 5.10 \* )
5.  $(p, g)$  是 ElGamal 体制的公开参数;
6.  $G$  是明文消息空间。  
( \* 其余部分与算法 8.3 的相同 \* )

另外, 由 Euler 准则(见 6.5.1 节的定理 6.13)我们知道  $g \in \text{QR}_p$ , 从而  $G = \text{QR}_p$  (读者可以根据 5.2.3 节的定理 5.2 推出)。于是, 对于选择的明文  $m_0, m_1 \in \text{QR}_p$ , 数  $g, y, c_1^*, c_2^*$  均为模  $p$  的二次剩余。因而, 所有二次剩余检验都将输出回答 YES, 例 14.1 中描述的二次剩余攻击将不再成功。

对明文空间是  $G = \text{QR}_p$  的规定不会影响消息的编码(在加密时)和解码(在解密时)。例如, 对任意的消息  $m < p$ , 如果  $m \in \text{QR}_p$ , 则已经完成; 如果  $m \notin \text{QR}_p$ , 则  $-m = p - m \in G$ 。由

$$(-1)^{(p-1)/2} = (-1)^q = (-1)^{\text{奇数}} = -1 \pmod{p}$$

有

$$(-m)^{(p-1)/2} = (-1)^{(p-1)/2} m^{(p-1)/2} = (-1)(-1) = 1 \pmod{p}$$

因此根据欧拉准则有  $-m \in \text{QR}_p = G$ 。

对于 ElGamal 体制的这一变形形式, Malice 现在面临着另一个判定问题。提交  $m_0, m_1$  要求加密后, 他收到询问密文  $(c_1^*, c_2^*)$ , 他可以由  $c_2^*$  计算

$$c_2^*/m_0 = \begin{cases} y^k \equiv g^{xk} \pmod{p} & 50\% \text{ 的概率} \\ y^k (m_1/m_0) \pmod{p} & 50\% \text{ 的概率} \end{cases}$$

注意在第一种情况中,

$$(g, y, c_1^*, c_2^*/m_0) = (g, g^x, g^k, g^{xk}) \pmod{p}$$

是一个 Diffie-Hellman 组, 但在第二种情况中就不是。所以 Malice 会思考

$$(g, y, c_1^*, c_2^*/m_0) \pmod{p} \text{ 是一个 DH 组吗?}$$

或者

$$(g, y, c_1^*, c_2^*/m_1) \pmod{p} \text{ 是一个 DH 组吗?}$$

也就是说, IND-CPA 游戏实际上是询问 Malice 要求回答  $G$  中的 DDH 问题(见 13.3.4.3 节的定义 13.1)。

如果 Malice 可以正确地回答  $G$  中的 DDH 问题, 那么给定询问密文对, 他当然可以正确指出被加密的明文, 即正确指出  $\mathcal{O}$  的硬币投掷结果。相反, 因为  $(g, y, c_1^*, c_2^*/m_0) \pmod{p}$  和  $(g, y, c_1^*, c_2^*/m_1) \pmod{p}$  是由  $g$  生成的随机组, 所以如果 Malice 能够正确指出被加密的明文, 那么他就能正确地回答  $G = \langle g \rangle$  中的 DDH 问题。这样, 对于使用算法 14.2 中公开参数的 ElGamal 体制, 它的 IND-CPA 安全性恰恰就是回答  $G$  中 DDH 问题的困难性(定理 14.2)。

在阿贝尔群的一般情况下(包括算法 14.2 中定义的  $G$ ), 我们不知道可以回答 DDH 问题的任何有效算法。这个困难性使 DDH 问题成为一个标准而且是广泛认可的困难问题假设。为了对这一问题进一步研究, 读者可以参考 Boneh 的研究报告[48]。

**假设 14.2 有限域中的判定 Diffie-Hellman 假设(DDH 假设)** 假设  $\mathcal{IG}$  是一个群实例生成器, 输入为  $1^k$ , 运行时间是关于  $k$  的多项式, 输出 (i)  $\text{desc}(G)$  (有限域上一个阿贝尔群  $G$  的描述), 其中  $|G| = k$ , (ii) 群的一个生成元  $g \in G$ 。

我们称  $\mathcal{IG}$  满足判定 Diffie-Hellman(DDH) 假设, 如果对于所有足够大的  $k$  和  $(\text{desc}(G), g) \leftarrow \mathcal{IG}(1^k)$ , 总体  $(g, g^a, g^b, g^{ab})$  和  $(g, g^a, g^b, g^c)$  是多项式不可区分的, 这里多项式不可区分的概念由定义 4.14(见 4.7 节)给出。

需要注意, 我们只在有限域的群中考虑 DDH 假设, 而不是一般的阿贝尔群, 因为由超奇异椭圆曲线上的点够成的群中的 DDH 问题是容易的(见 13.3.4.3 节)。

我们为 ElGamal 体制的变形建立了如下结论。

**定理 14.2** 使用算法 14.2 中公开参数的 ElGamal 体制是 IND-CPA 安全的, 当且仅当 DDH 假设成立。 □

### 14.3.6 基于 Rabin 比特的语义安全密码体制

在 Goldwasser 和 Micali 对具有语义安全性的密码体制的研究之后, 研究者们又提出了一些语义安全的密码体制, 并对 GM 体制进行了一些形式上的改进, 这包括 Blum 和 Micali 的方案[47]、Yao 的方案[305], 以及 Blum 和 Goldwasser 提出的一种高效方案[46]。

这些改进的主要思想是 CSPRB 生成器的概念(参阅前面的小节)。它是以一个  $k$  比特的随机种子为输入的程序,输出一个  $k'$  比特的数,  $t > 1$  是固定的。CSPRB 生成器的输出在下述意义下是一个高质量的生成器:如果  $k$  比特种子完全未知,那么输出的  $k'$  比特数与同长的真随机数在任何关于  $k$  的多项式时间内的统计测试下都不可区分。

现在,为了加密一条  $\ell$  比特的消息  $m$ ,发送者把  $m$  同 CSPRB 生成器的输出  $pr$  异或相加,这个生成器的输出以  $k$  比特的输入  $s$  为种子,异或相加  $m$  和输入  $k$  比特种子  $s$  时 CSPRB 的  $\ell$  比特输出  $ps$ ,把该结果以及用一个公钥对  $s$  的加密发送给接收者,即

$$(c_1, c_2) = (\mathcal{E}_{pk}(s), m \oplus ps) \quad (14.3.3)$$

合法的接收者(即公钥  $pk$  的主人)可以解密  $c_1$ ,得到种子  $s$ 。这样接收者可以由 CSPRB 再生成同样的  $\ell$  比特伪随机数  $ps$ ,用异或操作由  $c_2$  恢复出  $m$ 。

基于 CSPRB 生成器的加密方案比逐比特加密在效率上有很大的改进。一个  $\ell$  比特的明文消息现在扩展成  $(\ell + k)$  比特的密文消息,而不是逐比特加密时的  $\ell k$  比特。改进后的时间和空间复杂度与 RSA、Rabin 和 ElGamal 等教科书加密方案类似。

#### 14.3.6.1 基于 CSPRB 的加密方案的语义安全性

如果种子  $s$  是一个均匀随机的  $k$  比特串,而且基于分组的确定性加密算法  $\mathcal{E}_{pk}$  (安全参数为  $k$ ) 构成消息空间上的一个置换,那么式(14.3.3)中的第一个密文块  $c_1$  是从一个均匀分布的随机数置换得到的,所以也是均匀分布的。因此,它不会为攻击者提供有关明文的任何先验或后验信息。

RSA 加密算法是消息空间上的一个置换。如果  $N$  是一个 Blum 素数, Rabin 加密就可以构造  $QR_N$  上的一个置换,这一点已在定理 6.18(iv)(见 6.7 节)中说明。所以,这些算法都是  $\mathcal{E}_{pk}$  的良好候选。

进一步,用种子  $s$  生成的伪随机串  $ps$  起到了加密内部随机操作的作用,这归因于 CSPRB 生成器的强度。因此,  $m$  和  $ps$  的异或运算提供了  $m$  的语义安全加密。

对于由 Blum 和 Goldwasser 提出的基于 CSPRB 生成器的有效加密方案(BG 密码体制,[46]),式(14.3.3)中的  $c_1$  是  $s^{2^i} \pmod N$ , 其中  $i = \lfloor \frac{\log_2 m}{\log_2 \log_2 N} \rfloor + 1$ ; 伪随机比特串  $ps$  是由种子  $s$  使用逐分组形式的 BBS 伪随机生成器(9.3.1)生成的:每一个分组是  $s$  的  $2^j$  次幂模  $N$  ( $j = 1, 2, \dots, i-1$ ) 的  $\log_2 \log_2 N$  个低位比特。注意密文对的第一部分本质上是  $s$  的一个 Rabin 加密。

因为从 Rabin 密文同时提取明文的  $\log_2 \log_2 N$  个低位比特等价于分解  $N$  (回顾 9.3.1 节中的注释 9.1),所以 BG 体制的语义安全性可以量化为等价于  $N$  的分解。

### 14.4 语义安全性的不充分性

定义 14.1(以及性质 14.1)中引入的 IND-CPA 安全性(语义安全性)给我们这样一种直觉:给定明文对应的密文,任何多项式有界的攻击者都不能获得有关该明文的任何先验信息。然而,对明文安全性的这一保证只有当面对密文时攻击者是被动的情况下才有效,即攻击者所能做的只是搭线窃听。

在 8.6 节和 8.14 节中我们已经指出很多公钥密码体制对所谓的选择密文攻击 CCA 或 CCA2, 见 8.6 节中的定义 8.3) 都特别脆弱。在 CCA 和 CCA2 中, 攻击者(现在是 Malice)可以得到解密帮助, 即他可以在某种程度上控制“解密盒”, 因此即使他不知道解密密钥, 也可以对他选择的一些密文解密。我们已经将这种解密帮助看做是为了使给 Malice 的工作更容易而给他提供的一种“分析训练课程”。这些攻击模式, 尤其是 CCA2, 在公钥密码体制的很多应用中都是很实际的。例如, 某些协议可以要求对一个随机的询问执行解密操作以形成一种询问-应答机制。还有一个例子是加密后电子邮件的接收者可能会在随后的公开讨论中泄露明文。

很多公钥密码体制都对 CCA 或 CCA2 特别脆弱, 这是因为构成这些体制的基础一般都具有良好的代数性质。Malice 可以研究这些良好的代数性质, 通过某些巧妙的计算构造一条密文。有了解密帮助, 基于目标体制的良好代数特性, Malice 对选择密文进行的巧妙计算就能使他获得他本来得不到的消息。

在例 8.9 中, 我们看到 ElGamal 体制对 CCA2 攻击非常脆弱。这种攻击显然也可以应用到 IND-CPA 安全性的 ElGamal 体制变形上。同种类型的 CCA2 攻击也明显可以应用到基于 CSPRB 生成器的任何 IND-CPA 安全方案上(见 14.3.6 节); 在这类攻击中, 用  $c'_2 = r \oplus c_2$  替代式(14.3.3)中的  $c_2$ , 其中  $r$  是  $\ell$  比特的随机串, 与例 8.9 中随机数  $r$  的作用(盲化)相同。

例 14.2 将给出 GM 体制对 CCA2 的脆弱性。

**例 14.2** 假设 Malice 可以有条件地控制 Alice 的 GM 解密盒。这个条件相当“合理”: 如果对 Malice 所提交密文的解密结果看起来是随机的, 那么 Alice 就给 Malice 返回明文。

假设密文  $C = (c_1, c_2, \dots, c_\ell)$  是来自 Alice 同另一个人(不是 Malice)进行秘密通信时明文  $B = (b_1, b_2, \dots, b_\ell)$  的加密。但是 Malice 窃听到了  $C$ , 他想知道  $B$ 。现在, 他发送给 Alice 如下“巧妙计算的密文”:

$$C' = (yc_1, yc_2, \dots, yc_\ell) \pmod{N} \quad (14.4.1)$$

在这个攻击中, Malice 将利用下述良好的代数性质:

$$ab \pmod{N} \in \text{QR}_N \text{ 当且仅当 } \begin{cases} a \in \text{QR}_N & \text{且 } b \in \text{QR}_N \\ a \in \text{J}_N(1) \setminus \text{QR}_N & \text{且 } b \in \text{J}_N(1) \setminus \text{QR}_N \end{cases}$$

这一性质是应用欧拉准则(6.5.1 节的定理 6.13)的直接结果。

因为  $y \in \text{J}_N(1) \setminus \text{QR}_N$ , 我们可以将  $y$  看做是比特 1 的加密。这样, 在式(14.4.1)中“乘以  $y$ ”的攻击就会导致比特  $b_i$  的翻转,  $i = 1, 2, \dots, \ell$ , 也就是说, Alice 的解密结果是

$$B' = (b'_1, b'_2, \dots, b'_\ell).$$

$b'_i$  表示比特  $b_i$  的补,  $i = 1, 2, \dots, \ell$ 。

这一解密结果在 Alice 看来是随机的。所以 Alice 将  $B'$  返回给 Malice。这样, Malice 就找到了  $B$ 。

Malice 也可以使用“乘子”  $Y = (y_1, y_2, \dots, y_\ell)$  而不是  $(y, y, \dots, y)$ , 使得  $B'$  均匀随机(而不只是“看起来随机”), 这里  $Y$  是在 Alice 的公开密钥下对均匀随机  $\ell$  比特组  $Z = (z_1, z_2, \dots, z_\ell) \in {}_U\{0, 1\}^\ell$  的 GM 加密。很容易检验

$$B = (b'_1 \oplus z_1, b'_2 \oplus z_2, \dots, b'_\ell \oplus z_\ell)$$

□

在这个攻击中, Alice 为 Malice 提供了解密帮助的“预言服务”。注意预言服务未必是明确的。例 14.3 将说明不回答 Malice 的密文提问未必是一个好的策略。

**例 14.3** 假设现在 Alice 再也不会给 Malice 返回看起来随机的解密结果。对于加密后的消息  $C = (c_1, c_2, \dots, c_\ell)$  (如 Bob 发给 Alice), Malice 仍然可以逐比特恢复到明文。

例如, 为了得到  $c_1$  是 0 还是 1 对应的密文, Malice 可以发送给 Alice 一个加了密的问题(例如, 一个需要回答 YES/NO 的问题)。Malice 可以以通常的方式加密该问题的前半部分, 但是在下半部分加密中用  $c_1$  代替算法 14.1 的  $y$ 。

如果  $c_1 \in \text{QR}_N$ , 那么 Alice 只能正确解密前半部分。其余问题的解密会全是 0。这样她会问 Malice 为什么只发送一个没有完成的语句。此时 Malice 就知道了  $c_1$  是 0 对应的密文。另一方面, 如果 Alice 可以正确回答该问题, 那么 Malice 就知道  $c_1$  是一个非二次剩余, 因此是 1 对应的密文。

注意, 在这种攻击方法中, Malice 甚至可以对他的所有消息进行数字签名, 让 Alice 完全确认这些消息的真实来源。Malice 不会因为操作错误而受到控告!  $\square$

从这两种主动攻击的方法我们认识到 GM 体制毫无希望抵抗主动攻击。事实上, IND-CPA 模型下的安全性定义太弱了。

## 14.5 超越语义安全性

将“完全或无”意义下的安全性(见 8.2 节性质 8.2)提升到 IND-CPA 安全性(定义 14.1)是我们强化安全性概念的第一步。

在 14.4 节中我们看到 IND-CPA 意义下的安全性概念在应用中还不够好, 在这些应用中, 用户可能被欺骗而提供解密模式下的预言服务。在密码系统的应用中, 要求天真的用户总是保持警醒不提供解密预言服务的确不实际。所以, 我们需要更强的安全性概念。

强化安全性概念的下一步是考虑另一个攻击模型, 称为不可区分选择密文攻击(IND-CCA)。在该模型中, 我们可以进一步降低 Malice 攻破目标密码体制的难度: 除了在 CPA 游戏(协议 14.1)中可以获得的加密帮助以外, 我们进一步允许 Malice 获得解密模式下的有条件帮助。对 IND-CCA 模型的形式化处理基于 Naor 和 Yung[212]给出的游戏。该游戏称为“午餐攻击”或“冷漠选择密文攻击”。

### 14.5.1 抗击选择密文攻击的安全性

午餐攻击描述了机构中其他员工不在的情况下(如午餐时间)Malice 的真实场景。他询问该机构中的解密机制, 希望通过与解密盒的交互可以为他提供一种“密码分析训练课程”, 使他将来分析该机构密码体制时更有经验。因为午餐持续时间短, 所以 Malice 没有足够的时间准备他的提问密文, 使其在某个函数下与解密盒的回答相关。因此, 午餐时间他提问的所有密文都是在午餐前准备好的。

这个真实场景也可以由一个攻击游戏模型化。该游戏的参与者与 IND-CPA 攻击游戏(协议 14.1)的参与者相同: Malice, 他可能是一个对机构不满的员工, 以及预言机  $\mathcal{O}$ , 现在是该机构

的解密(和加密)机制。我们把这个游戏称为不可区分的选择密文攻击 (IND-CCA)。协议 14.3 具体描述了这个新游戏。

### 协议 14.3 “午餐攻击”(非适应性不可区分选择密文攻击)

假定

- i) 如协议 14.1, Malice 和  $\mathcal{O}$  商定了一个目标密码体制  $\mathcal{E}$ ,  $\mathcal{O}$  选定了  $\mathcal{E}$  的一个加密密钥;
- ii) Malice 在午餐前准备好了一些密文消息。
  1. Malice 向  $\mathcal{O}$  发送一条准备好的密文消息  $c \in \mathcal{C}$ ;
  2.  $\mathcal{O}$  解密  $c$ , 返回解密结果给 Malice;
 

( \* 密文  $c$  称为选择密文或冷漠选择密文; 把解密结果返回给 Malice 被认为是给他提供“分析训练课程”; Malice 可以要求多次重复“训练课程”, 直到如他所愿; 因为午餐时间很短, 他可以考虑使用一个程序加速“训练课程” \* )
  3. 一旦 Malice 对“解密训练课程”感到满意, 他就要求  $\mathcal{O}$  进行协议 14.1 中的 CPA 游戏。
 

( \* 在 CPA 游戏的这个实例中, 选择的明文消息  $m_0$  和  $m_1$  称为适应性选择明文; 也就是说, 这两条消息可以是第 1 步和第 2 步中提供的整个“解密训练课程”历史的某个函数; 因此 Malice 的“寻找阶段”正好在该协议的开始部分开始, 并且在收到询问密文  $c^* \in \mathcal{C}$  时结束,  $c^*$  是等可能的  $m_0, m_1 \in \mathcal{M}$  之一的加密 \* )

( \* 因为对明文的处理比对密文的处理相对容易得多, 我们可以合理地假设即使在很短的午餐时间, Malice 也能够计算适应性选择明文消息 \* )

( \* 现在, “午餐时间”结束; Malice 应该回答 0 或 1, 作为他在 CPA 游戏中对  $\mathcal{O}$  的硬币投掷的有根据猜测; 但是, 除非 Malice 做出了回答, 否则即使游戏结束, 他也还是处于“猜测阶段”。\* )

乍一看, 人们可能会认为这个午餐攻击游戏不能模型化真实的攻击场景。谁会那么好那么天真地扮演解密盒的作用, 回答 Malice 的解密提问呢? 我们将从四个方面回答这个问题。

- 在密码学的很多应用中(尤其是在密码协议中), 通常要求用户(协议的一个参与者)一旦收到询问消息, 就用她的私钥执行解密操作, 并将解密结果返回。这就是所谓的询问-应答机制(见第 2 章和第 11 章)。
- 我们可能要接受一个生活现实: 很多用户恰恰就是不可救药地天真, 我们很难要求或教育他们在有坏人实施诡计时保持高度的警觉。事实上, 我们说更强的密码体制和安全性概念就是为这些天真用户设计的并不为过。
- Malice 可以将解密提问嵌入到正常而且看起来无害的通信中, 这样做他可能得到他的提问的隐含回答。例 9.2 和例 14.3 就提供了生动的主动攻击, 它们看起来都如此无害。把这样的攻击同合法的安全通信区分开来往往是非常困难的。不回答任何问题(加了密的问题或答案)并不是解决主动攻击的好方案, 而只是自我否定安全通信技术的优点。
- Malice 甚至可以利用潜信道, 例如我们在 12.5.4 节中介绍过的定时分析攻击, 以时间延迟差的形式回答 Malice 的问题。



对待 Malice 的正确态度就是直截了当地面对他,并为他提供他需要的“密码分析训练课程”。训练课程可以是加密也可以是解密,可以是整个数据块也可以是单个比特。我们的策略是设计强的密码体制,使得即使我们按要求提供“密码分析训练课程”,也不会有助于 Malice 攻破目标密码体制。

根据与 14.2 节推导 CPA 游戏(协议 14.1)中 Malice 攻破目标密码体制的优势相同的道理,我们可以类似地得到 Malice 在午餐攻击中的优势。优势的公式与式(14.2.3)非常类似,但是现在我们要在 Malice 的输入中加上选择密文分析训练课程的整个历史。令 Hist-CCA 表示这个历史,则 Malice 的优势为

$$\text{Porb}[1 \leftarrow \text{Malice}(c^*, m_0, m_1, \text{Hist-CCA}) \mid c^* = \mathcal{E}_k(m_1)] = \frac{1}{2} + \text{Adv} \quad (14.5.1)$$

现在,我们得到了一个在 IND-CPA 概念基础上加强的新安全性概念。

**定义 14.2 对不可区分选择密文攻击的安全性(IND-CCA 安全性)** 一个参数为  $k$  的密码体制被称为对于不可区分选择密文攻击是安全的(IND-CCA 安全的),如果任何多项式有界的攻击者进行协议 14.3 中的攻击游戏之后,式(14.5.1)给出的优势 Adv 公式都是关于  $k$  的可忽略量。

因为在午餐攻击中,对 Malice 的解密帮助(或“密码分析训练课程”)是在协议 14.1 中 IND-CPA 游戏的基础上提供的,所以新的攻击游戏必然会降低 Malice 分析任务的难度。因此我们可以认为 IND-CPA 安全的某些密码体制可能不是 IND-CCA 安全的。

本章已经介绍的 IND-CPA 安全的密码体制都没有被证明是 IND-CCA 安全的。它们中有一个明显不安全!那就是我们在 14.3.6 节中介绍的 Blum 和 Goldwasser 基于 CSPRB 生成器的有效密码体制。

**例 14.4** 为了在午餐攻击模式下攻击 BG 体制, Malice 需要做一个选择密文提问  $(c, m)$ , 其中选择  $c$  为一个模  $N$  二次剩余,  $|m| = \lfloor \log_2 \log_2 N \rfloor$ 。观察 14.3.6 节中描述的 BG 体制,我们知道对 Malice 的回答将是以下解密结果:

$$m \oplus "c \text{ 模 } N \text{ 的一个平方根的 } \lfloor \log_2 \log_2 N \rfloor \text{ 个低位比特}."$$

将该回答与  $m$  逐比特异或, Malice 就得到了  $c$  模  $N$  的一个平方根的  $\lfloor \log_2 \log_2 N \rfloor$  个低位比特。回忆选择的  $c$  为一个模  $N$  二次剩余。回顾注释 9.1(见 9.3.1 节),为 Malice 提供  $c$  的一个平方根的  $\lfloor \log_2 \log_2 N \rfloor$  个低位比特可以使他在概率多项式时间内分解  $N$ !  $\square$

例 14.4 表明确实就是这个为 Malice 提供的密码分析训练课程使他能够攻破该体制。结果是如此的严峻和彻底,不仅仅是一条密文的泄露,而且是整个体制的毁灭。

我们也意识到 BG 体制 IND-CPA 安全的根基恰好也是该体制对 IND-CCA 不安全的真正原因。这与讨论 Rabin 体制在“完全或无”意义下安全性时的情况类似(见 8.11 节的定理 8.2)。

Naor 和 Yung 提出了一个抗击 IND-CCA 可证明安全的密码体制[212]。在该体制中,明文在两个不同公钥下逐比特加密成两条密文消息。加密算法包括一个非交互式零知识(NIZK)证明过程,它使明文消息的发送者能够证明这两条密文确实是同一明文比特分别在两个公钥下的加密(将加密算法看成是形成一个 NP 问题,以明文和算法的随机输入作为该 NP 问题的证言,参考 4.8.1 节中的讨论)。该证明在解密时将得到验证(如在午餐攻击中由  $\mathcal{O}$  完成)。如

果解密时通过验证,则表明发送者已经知道(如午餐攻击中 Malice 已经知道)加密成该密文对的明文。因此午餐时间为 Malice 提供服务并不会给他提供新知识,不会使他的分析工作更容易。由于实现对一个加密(解密)算法的 NIZK 证明(验证)以及逐比特加解密的花销太大,所以 Noar 和 Yung 的体制[212]不适于实际应用。

午餐攻击是一个非常受限的攻击模型, Malice 只能在一个很短的时间段内获得提供给他解密服务,就好像“午餐”后解密盒被永久地关闭一样,或者,即使在第二天的“午餐时间”, Malice 也不会再次攻击似的。这不合理也不现实。在实际中,天真的用户会永远保持天真, Malice 无疑还会打回来,甚至可能就在下午茶休息时间! 因此在 IND-CCA 模型下的安全性概念还是不够强。我们需要更强的安全性概念。

### 14.5.2 抗击适应性选择密文攻击的安全性

我们强化安全性概念的更进一步是考虑称为不可区分适应性选择密文攻击(IND-CCA2)的模型。Rackoff 和 Simon 最初提出了这种更强的攻击模型[243]。

在这个模型中,我们在午餐攻击的基础上进一步降低 Malice 攻击密码体制的难度。在午餐攻击中(协议 14.3),解密帮助(或“密码分析训练课程”)是有条件的,一旦 Malice 提交两条适应性选择明文消息该帮助就停止,也就是说,一旦 IND-CPA 攻击(协议 14.1)游戏开始,午餐攻击就结束,而且从此再也不能得到解密帮助。

在新的攻击模型中,我们去掉只能在短时间内得到解密帮助这个不现实的条件,午餐攻击中的这个条件在某种程度上是人为加上的。现在,对 Malice 的解密帮助在午餐攻击前后都永远可得。我们可以把这种攻击情景设想为一个延长的午餐攻击。因此,我们命名这个新的攻击模型为**凌晨攻击**。这一名称表现了 Malice 的真实情景,他还是一个对机构不满的员工,熬通宵使用机构的解密机制。注意凌晨攻击与所谓的午夜攻击不同,午夜攻击经常以另一种名称出现——午餐攻击;可能与在午餐攻击中的情况一样,人们认为机构的安全警卫在午夜时间会非常准时地吃饭。

因为现在 Malice 有足够的时间而不被发觉,所以他当然可以以更复杂更有意义的方式使用解密盒。除了他在午餐攻击(事实上是在午夜)中可以做的以外,即使用从“解密训练课程”中收集到的信息以及接下来得到的相应询问密文,适应性地选择明文提问,现在 Malice 收到询问密文之后仍然可以提交适应性选择密文。因此,这些适应性选择密文消息可以在某种程度上与询问密文相关,因为与询问密文对应的明文是他自己选择的。当然,解密盒是足够聪明的,它不会为 Malice 提供询问密文的解密! 这是惟一的限制,当然也是合理的。如果没有这个限制, Malice 只需要求解密盒解密询问密文,我们也就没有这样有趣的游戏可以做了! 解密盒又必须足够笨,它会解密任何与询问密文有直接关系的密文! 对密文的任何微小改变,譬如乘以 2,或者加 1,都一定能得到解密服务!

我们对新攻击模型的描述在协议 14.4 中。

---

#### 协议 14.4 “凌晨攻击”(不可区分适应性选择密文攻击)

假定

和协议 14.1 一样, Malice 和预言机  $\mathcal{O}$  商定了一个目标密码体制  $\mathcal{E}$ ,  $\mathcal{O}$  选定了一个加密密钥。

1. Malice 和  $\mathcal{O}$  做协议 14.3 中的午餐攻击游戏;

( \* 在午餐攻击游戏的这个实例中, Malice 的“寻找阶段”与协议 14.3 中的相同, 在他收到询问密文  $c^* \in C$  后就结束,  $c^*$  是等可能的选择明文  $m_0, m_1 \in M$  之一的加密; 但是在该实例中, 允许 Malice 扩展他的“猜测阶段”, 扩展的“猜测阶段”如下 \* )

2. Malice 进一步计算密文  $c' \in C$ , 将其提交给  $\mathcal{O}$  解密;

( \* 密文  $c'$  称为适应性选择密文或询问后选择密文; 相反, 在午餐攻击游戏中选择的密文(协议 14.3)也被称为询问前选择密文; 可以认为步骤 2 为 Malice 提供了在午餐攻击基础上扩展的“解密训练课程”; Malice 可以要求“扩展的训练课程”重复多次, 直到他满意为止 \* )

( \* 约定  $c' \neq c^*$ , 即不允许 Malice 将询问密文发回要求解密 \* )

3. 一旦 Malice 对“扩展解密训练课程”感到满意, 他就必须回答 0 或 1 作为他对  $\mathcal{O}$  的硬币投掷的有根据猜测。

又根据 14.2 节推导 Malice 在 IND-CPA 游戏(协议 14.1)中攻破目标体制的优势时相同的道理, 我们可以类似地推出 Malice 在凌晨时间攻击游戏中攻破目标体制的优势。该优势公式又和式(14.2.3)非常类似, 但是现在我们必须要在 Malice 的输入中加上两次分析训练课程整个历史, 这两次分别是询问前的 CCA 和询问后的 CCA 或“扩展的 CCA”。用 Hist-CCA2 表示这个历史, 则 Malice 的优势是:

$$\text{Prob}[1 \leftarrow \text{Malice}(c^*, m_0, m_1, \text{Hist-CCA2}) \mid c^* = \mathcal{E}_k(m_1)] = \frac{1}{2} + \text{Adv} \quad (14.5.2)$$

现在我们得到了一个在 IND-CCA 基础上进一步加强的新安全性概念。

**定义 14.3 抗击不可区分适应性选择密文攻击的安全性(IND-CCA2 安全性)** 参数为  $k$  的密码体制称为对不可区分适应性选择密文攻击是安全的(IND-CCA2 安全的), 条件是任何多项式有界的攻击者进行协议 14.4 中的攻击游戏之后, 式(14.5.2)给出的优势 Adv 公式都是关于  $k$  的可忽略量。

我们将目前已经介绍过的各种 IND 攻击总结在图 14.1 中。

因为在凌晨攻击游戏中, “扩展的”解密帮助(或“扩展的密码分析训练课程”)是在午餐攻击之后提供的, 所以新的攻击游戏必然在午餐攻击的基础上降低了 Malice 攻破目标体制的难度。因此我们预想某些 IND-CCA 安全的体制可能不再是 IND-CCA2 安全的。事实上, 除了我们在算法 10.6 中介绍的 RSA-OAEP 之外, 本书目前已经介绍的所有体制都不是可证明 IND-CCA2 安全的。我们给出了很多在“完全或无”意义下对 CCA2 不安全的密码体制, 因此也不是 IND-CCA2 安全的(参考例 8.4, 8.6, 8.8, 9.2, 14.2, 14.3)。

介绍了 IND-CCA2 的概念之后, Rackoff 和 Simon 也提出了一个基于 NIZK 证明的 IND-CCA2 安全的体制。但是, 他们考虑的是具有特定证明者的 NIZK 证明。在他们 IND-CCA2 安全的体制中, 不仅接收者有一对公私钥, 而且发送者也有这样的一对密钥。发送者的公钥是由一个公钥证书设施(参考 13.2 节中的方法)发给的。发送者不仅像通常一样用接收者的公钥加密, 而且在构造 NIZK 证明时使用他自己的私钥, 于是密文的接收者可以用发送者的公钥进行验证。NIZK 验证通过说明该密文确实是这个发送者(证明者)生成的, 因此, 将对应的明文返回给发

送者不会给他提供有利于攻破目标体制的任何信息。Rackoff 和 Simon 的 IND-CCA2 安全体制也是逐比特操作的。

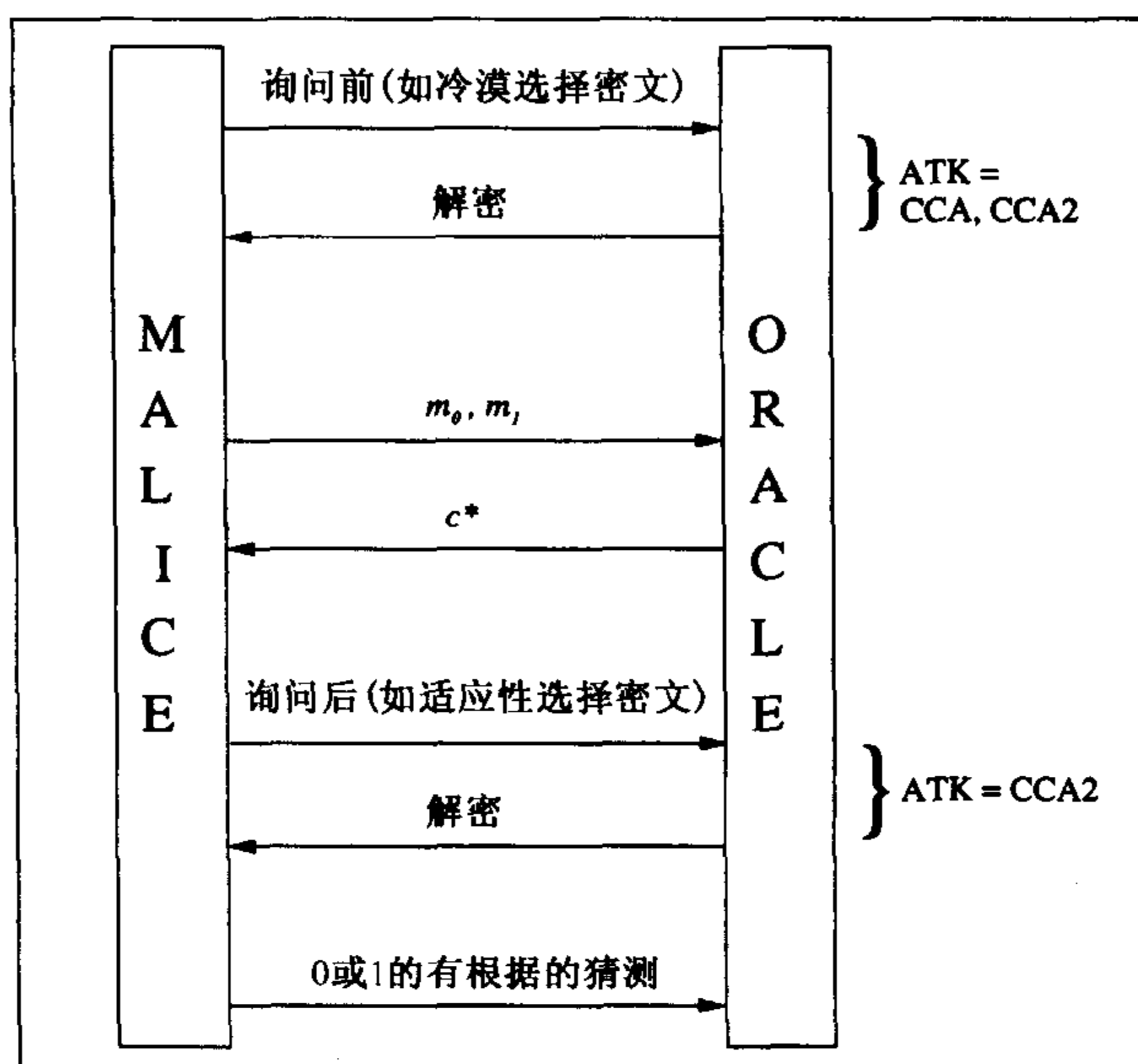


图 14.1 不可区分攻击游戏总结

### 14.5.3 不可展密码学

不可展(NM)密码学[101]将从计算的方面加强公钥密码体制的安全性定义。NM 是一个重要的要求,它使得 Malice 很不容易以一种有意义的可控方式通过修改密文而修改相应的明文。Dolev 等人使用一个合同竞标例子很好地说明了这一要求的重要性。

假设多个建筑公司接受地方政府的邀请竞标建筑一座新的小学。积极促进电子操作的政府公开自己的公钥  $E$ , 该公钥用于加密标书, 并建立了一个电子邮件地址 `e-gov@bid.for.it.gov` 用于接收加密的数据。公司 A 的标价定为 1 500 000 美元, 并将  $E(1\,500\,000)$  发送至 `e-gov@bid.for.it.gov`。但是, 该邮件被 CheapSub 的头目 Malice 截获, CheapSub 是只有一个人的公司, 专门卖子合同给某些廉价建筑公司。如果电子政府使用的加密算法是可展的, 那么 Malice 就可以将  $E(1\,500\,000)$  改为  $E(15\,000\,000)$ 。这样, Malice 自己的标书就会有更大的机会中标。

最简单的一种可展加密算法是一次一密。在第三部分我们还看到所有最基本、最常用的公钥加密函数都是可展的。

与不可区分性的各种攻击情况不同, 它们的攻击问题是判定问题, 而可展性攻击的则是一个计算问题。我们在协议 14.5 中描述这个问题。

#### 协议 14.5 选择明文模式下的可展性攻击

假定

和协议 14.1 一样, Malice 和预言机  $\mathcal{O}$  商定了一个目标密码体制  $\mathcal{E}$ ,  $\mathcal{O}$  选定了一个加密密钥  $pk$ 。

Malice 和  $\mathcal{O}$  进行下面的游戏:

1. Malice 发送给  $\mathcal{O}$ :  $\mathbf{v}, \text{desc}(\mathbf{v})$ , 这里  $\mathbf{v}$  是一个向量, 包含多条明文,  $\text{desc}(\mathbf{v})$  是对  $\mathbf{v}$  中明文分布的描述;
2.  $\mathcal{O}$  生成一个有效的询问密文  $c^* = \mathcal{E}_{pk}(\alpha)$ ,  $\alpha$  的生成服从  $\mathbf{v}$  中明文的分布,  $\mathcal{O}$  将  $c^*$  发送给 Malice;
3. 一旦收到  $c^*$ , Malice 必须输出一个“有意义的”PPT 可计算的关系  $R$  和另一个有效的密文  $c' = \mathcal{E}_{pk}(\beta)$ , 使得  $R(\alpha, \beta) = 1$  成立。

在这个可展性攻击中, 因为给定询问密文  $c^*$ , Malice 的目标不是得到有关目标明文  $\alpha$  的信息, 他根本不需要知道  $\alpha$ 。但是, 为了攻击成功, Malice 必须输出一个有意义的关系  $R$  将  $c^*$  和  $c'$  的解密结果联系起来。

Malice 的成功也可以表示成优势的形式。在 [101] 中, 作者使用了零知识仿真<sup>①</sup>的思想表示这种优势。首先, 对于还是 PPT 算法的 Malice 给定  $c^* = \mathcal{E}_{pk}(\alpha)$ , 他以某个概率输出  $(\mathcal{E}_{pk}(\beta), R)$ 。然后, 一个模拟器, 我们记做 ZK-Sim, 它是一个 PPT 算法, 它不知道  $c^*$ , 但是也要以某个概率输出一条密文  $\bar{c}$  (ZK-Sim 甚至可以忽略加密算法和公钥)。Malice 进行可展性攻击的优势是下列概率差:

$$\begin{aligned} \text{NM-Adv} = & \text{Prob}[(\mathcal{E}_{pk}(\beta), R) \leftarrow \text{Malice}(\mathcal{E}_{pk}(\alpha), pk, \text{desc}(\mathbf{v}))] - \\ & \text{Prob}[(\bar{c}, R) \leftarrow \text{ZK-Sim}] \end{aligned} \quad (14.5.3)$$

安全参数为  $k$  的密码体制  $\mathcal{E}_{pk}()$  称为是不可展的, 如果对所有 PPT 可计算的关系  $R$  和所有 PPT 攻击者 (即 Malice 以及类似的攻击者), NM-Adv 都是关于  $k$  的可忽略函数。在 [101] 中, 该安全性概念被称为“在选择明文攻击下关于关系的语义安全性”。因此我们称之为 NM-CPA。NM-CPA 直观上达到了下述希望达到的安全质量。

**性质 14.2 NM-CPA 安全性** 给定从一个 NM-CPA 安全的密码体制得到的密文, Malice 对该体制进行可展性攻击, 那么他的优势不会比没有该密文时的攻击 (即仿真攻击) 优势以任何 PPT 内可以辨别的方式有所增加。

既然提供密文不会降低攻击难度, 那么提供“密码分析训练课程”也不会。与 IND-CCA 和 IND-CCA2 的情况类似, 可展性攻击的难度也可以在午餐攻击和凌晨攻击模式下降低。在午餐攻击模式下的可展性攻击中, Malice 可以将多个选择密文发送给  $\mathcal{O}$  解密, 但是一旦 Malice 要求得到询问密文  $c^*$ , 该服务就停止。在凌晨攻击模式下的可展性攻击中, Malice 得到询问密文  $c^*$  后解密服务不会停止。当然, 和我们在凌晨攻击游戏中规定的一样, 不允许 Malice 向  $\mathcal{O}$  发送回  $c^*$  要求解密。

这样, 我们就有了 NM-CCA 和 NM-CCA2 的概念。

因为涉及到这些问题的计算本质, 这里我们就不给出这些 NM 安全性概念的严格形式化处理。详细情况, 感兴趣的读者可以参阅 [101]。非常合理地, 我可以预料到 NM 安全性概念的形式化将涉及到某些 IND 安全性不会涉及到的复杂性。例如, 与判定问题的情况不同, 在判定问题中我们不需要注意明文消息空间的大小 (它甚至可以是 2, 例如 GM 体制中的情况), 而 NM 安全性的形式化必须规定密文消息空间足够大, 这样关系  $R$  才不会退化为一个平凡问题。

<sup>①</sup> 在后面的章节中, 我们将学习零知识证明及其多项式时间的仿真。



在[20]中,作者给出了一些稍微不同的 NM 安全性的形式化定义,它们基于我们介绍过的各种类似于 IND 攻击的游戏。由于他们的攻击与我们为定义 IND 安全概念而引入的攻击游戏很相似,读者可能会发现[20]中的处理更易于理解。

尽管如此,我们对 NM 安全性概念的描述已经足以精确地表达它的思想。很快我们就会看到大多数直接应用陷门单向函数的教科书加密算法都是极易展的。正如在第 9 章看到的,对于所有通用公钥密码体制使用的普通(陷门)单向函数,我们都可以在利用某些部分信息预言机(如“奇偶预言机”或者说“半阶数预言机”)的基础上求逆;而这些求逆方法的原理恰恰就是对未知明文消息进行可展性攻击。例如,对 RSA 的  $c = m^e \pmod{N}$  来说, Malice 知道如果对密文  $c$  乘以  $2^e \pmod{N}$ , 那么未知的明文  $m$  就会乘以 2。

Dolev 等提出了一个可证明 NM-CCA2 安全的公钥加密方案[101]。该方案使用多个公/私钥对,逐比特加密明文。对每个明文比特的加密也都包含一个 NIZK 证明。

#### 14.5.4 不可区分性与不可展性的关系

不可展安全性无疑是非常重要的。但是,由不可展问题的计算本质,对它们进行形式化处理非常困难。因此,设计一个密码体制并证实它具有不可展安全性也非常困难。

幸运的是,研究者建立了很多不可展安全性和不可区分安全性之间的重要关系,而且,在 CCA2 模型下,人们发现最有用的安全性概念——不可展性——与不可区分性等价。因为人们已经很好地建立了对 IND-CCA2 的形式化处理,所以我们可以证明 IND-CCA2 下的安全性获得在 NM-CCA2 模式下的可证明安全性。

对安全性概念之间关系的形式化证明可以通过构造一个多项式时间的归约算法得到。当把对密码体制的各种攻击联系起来时,这种归约(算法)将目标攻击问题(称为“目标攻击”)归约到另一个攻击(称为“来源攻击”)。如果一个成功构造的归约是 PPT 算法,那么基于“来源攻击”的成功“目标攻击”就可以成功,而且进行“目标攻击”的代价是关于进行“来源攻击”代价的一个多项式。

因为对密码体制的攻击总是基于某些适当的假设和要求(例如,为了 CCA2 攻击者能正确工作,应该允许其进行询问前和询问后的密码分析训练课),所以归约算法必须满足攻击者的这些必要假设和环境要求。构造归约时我们经常要用到一个特殊的代理,称为 **Simon 仿真器**。Simon 通过模拟攻击者的工作环境来满足攻击者的所有假设和合理要求。

有时,Simon 自己就变成了“目标攻击”的成功攻击者,因为他与“来源攻击”的攻击者交互后,该攻击者教会了他。在这样的归约中,Simon 妥善安排攻击者和加/解密预言机之间进行的两个攻击游戏。一方面,Simon 通过模拟“来源攻击”的环境(即假装成一个加/解密预言机面对攻击者)与攻击者进行“来源攻击”游戏。另一方面,Simon 与加/解密预言机进行“目标攻击”游戏,此时他是一个攻击者。在这样的情况下,我们可以认为“来源攻击”的攻击者正在教 Simon 进行“目标攻击”。图 14.2 和图 14.3 给出了 Simon 导演的一场精心安排。

现在我们可以陈述和证明一些有用的关系了。

##### 14.5.4.1 不可展性蕴涵不可区分性

假设 ATK 表示 CPA、CCA 或 CCA2。我们要证明如果一个公钥密码体制是 NM-ATK 安全的,那么它一定是 IND-ATK 安全的。



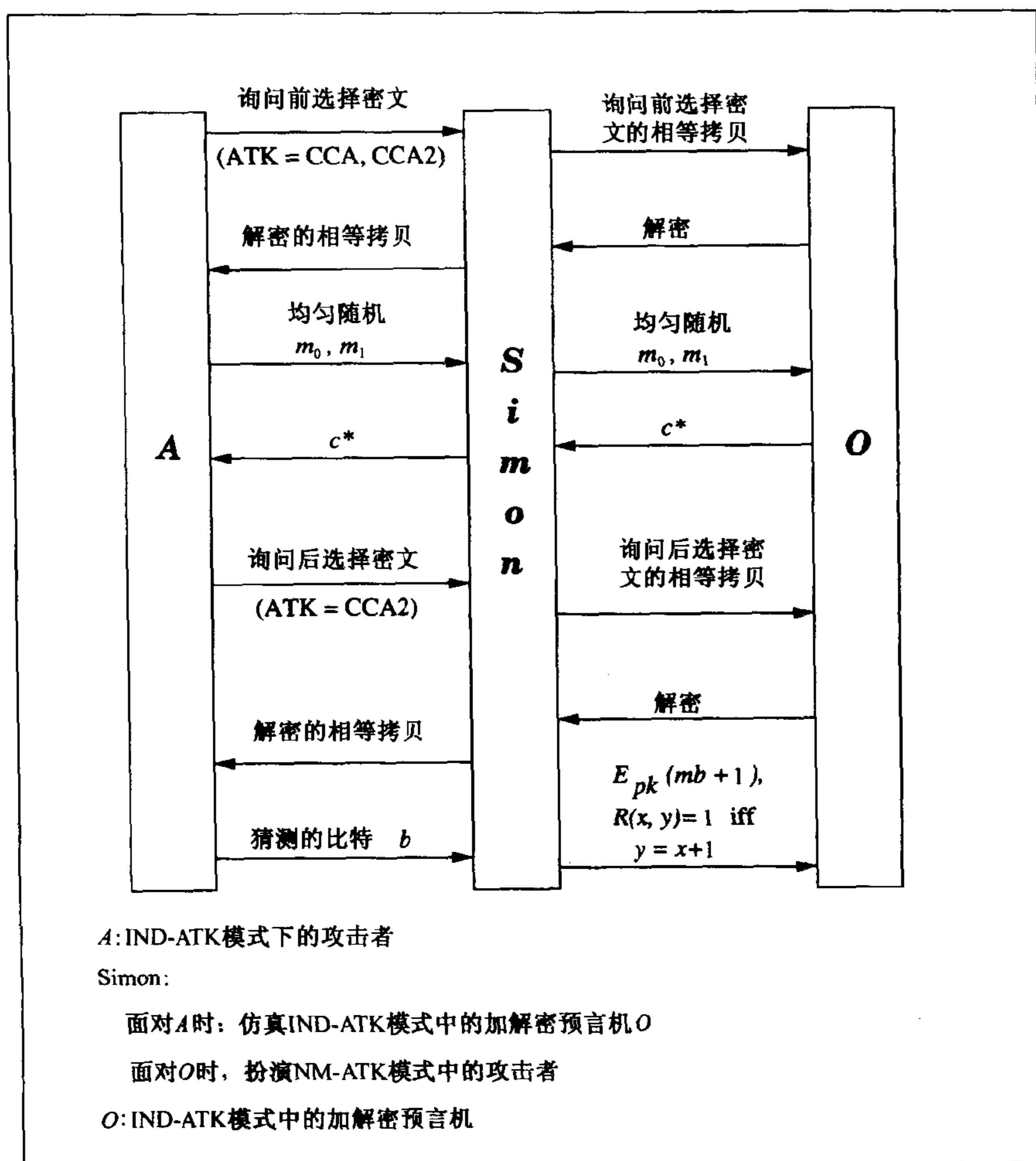


图 14.2 从 NM 攻击到 IND 攻击的归约

**定理 14.3** 如果一个公钥密码体制是 NM-ATK 安全的,那么它一定也是 IND-ATK 安全的。

**证明** 我们可以通过证明如下结论来证明该定理:如果一个公钥体制 $\mathcal{E}_{pk}$ 不是 IND-ATK 安全的,那么它一定不是 NM-ATK 安全的。

假设 $\mathcal{E}_{pk}$ 不是 IND-ATK 安全的,那么就有一个 PPT 攻击者 $\mathcal{A}$ 可以以一个不可忽略的优势 $\text{Adv}(\mathcal{A})$ 在 IND-ATK 模式下攻破 $\mathcal{E}_{pk}$ 。我们令 Simon 模拟器使用 $\mathcal{A}$ 构造一个归约,在 NM-ATK 模式下攻破 $\mathcal{E}_{pk}$ 。

Simon 协调两个攻击游戏。一个游戏是 IND-ATK(即协议 14.1、14.3、14.4 中的任何一个),其中 Simon 充当 $\mathcal{O}$ 的角色,与充当 Malice 的 $\mathcal{A}$ 交互。另一个游戏是 NM-ATK(即协议 14.5 形式的 ATK),其中 Simon 充当 Malice 的角色与加密预言机(和解密预言机,如果 $\text{ATK} \in \{\text{CCA}, \text{CCA2}\}$ ) $\mathcal{O}$ 交互。图 14.2 给出了最一般情况 $\text{ATK} = \text{CCA2}$ 时的归约。在其他两种 ATK 情况下,某些交互过程可以省略。

注意在可展性攻击中(即图 14.2 中右边的交互),选择明文分布均匀,因此 $\mathcal{O}$ 必须随机加密所选择明文中的一条。

$\mathcal{A}$ 的“有根据猜测”是  $b \in \{0,1\}$ 。Simon 可以输出  $c' = \mathcal{E}_{pk}(m_b + 1)$ 和关系  $R(x, y) = 1$ , 当且仅当对明文空间中的所有  $x$  都有  $y = x + 1$  时。显然, 由于  $\mathcal{A}$  是 PPT 的, 所以 Simon 也可以在多项式时间内输出这个正确的可展性结果。

因为  $\mathcal{A}$  以优势  $\text{Adv}(\mathcal{A})$  正确回答  $b$ , 所以有

$$\text{NM-Adv}(\text{Simon}) = \text{Adv}(\mathcal{A}) - \text{Prob}[(\bar{c}, R) \leftarrow \text{ZK-Sim}]$$

注意 ZK-Sim 不能访问询问密文  $c^*$ , 因此没有利用  $\mathcal{A}$ ; 于是对于模拟输出的、与满足  $R$  的明文相对应的密文  $\bar{c}$ ,  $\text{Prob}[(\bar{c}, R) \leftarrow \text{ZK-Sim}]$  必然是可忽略的。所以, 正如我们所愿,  $\text{NM-Adv}(\text{Simon})$  不可忽略。□

回忆一下, 我们介绍过了很多对不同密码体制在不同 IND-ATK 模式下的攻击。由定理 14.3, 我们知道这些体制在 NM-ATK 模式下也是不安全的。

我们知道存在 IND-CPA(IND-CCA)安全但不是 NM-CPA(NM-CCA)安全的密码体制。这些都可以在[20]中找到。

在[20]中已经得到的 NM 和 IND 安全性的关系中, 下面这个关系是最重要的。

#### 14.5.4.2 在适应性选择密文攻击下不可区分性蕴涵不可展性

在  $\text{ATK} = \text{CCA2}$  的情况下, 定理 14.3 的逆陈述也成立。

**定理 14.4** 公钥密码体制是 NM-CCA2 安全的当且仅当它是 IND-CCA2 安全的。

**证明** 因为在定理 14.3 中我们证明了  $\text{NM-CCA2} \Rightarrow \text{IND-CCA2}$ , 所以只要证明相反方向  $\text{IND-CCA2} \Rightarrow \text{NM-CCA2}$  即可。我们可以证明如果一个公钥密码体制  $\mathcal{E}_{pk}$  不是 NM-CCA2 安全的, 那么它也一定不是 IND-CCA2 安全的。

假设  $\mathcal{E}_{pk}$  不是 NM-CCA2 安全的, 则有一个 PPT 攻击者  $\mathcal{A}$ , 他可以以一个不可忽略的优势  $\text{Adv}(\mathcal{A})$  在 NM-CCA2 模型下攻破  $\mathcal{E}_{pk}$ 。令 Simon 模拟器使用  $\mathcal{A}$  构造一个归约在 IND-CCA2 模型下攻破  $\mathcal{E}_{pk}$ 。

图 14.3 给出了 Simon 构造的归约。注意该归约成为可能的原因恰恰就是可展性攻击者  $\mathcal{A}$  输出的密文  $c'$  与询问密文  $c^*$  不同, 这样这两个游戏的安排者, 也就是在 IND-CCA2 游戏中扮演 Malice 的 Simon, 可以将  $c'$  当做询问后选择的密文发送给  $\mathcal{O}$  解密。由解密结果, Simon 可以验证明文之间的关系(该关系由  $\mathcal{A}$  给出), 从而确定询问比特  $b$ 。

显然, 因为  $\mathcal{A}$  是 PPT 的, 所以 Simon 安排的这两个游戏也能在多项式时间内结束, 而且因为  $\mathcal{A}$  的优势不可忽略, 所以 Simon 的优势也不可忽略。□

图 14.4 总结了目前我们介绍过的安全性概念之间的已知关系。我们没有证明不蕴涵的情况(由  $\nrightarrow$  表示)。详细情况, 感兴趣的读者可以参阅[20]。

定理 14.4 告诉我们, 对于公钥加密体制, 只需要考虑 IND-CCA2, 它比 NM-CA2 容易处理。而且, 因为 IND-CCA2 和 NM-CCA2 等价, 所以人们已经普遍认可 IND-CCA2 就是通常用途公钥加密算法的正确安全性定义。

下一章我们将介绍两个实际中有效的公钥密码体制, 它们是可证明 IND-CCA2 安全的。

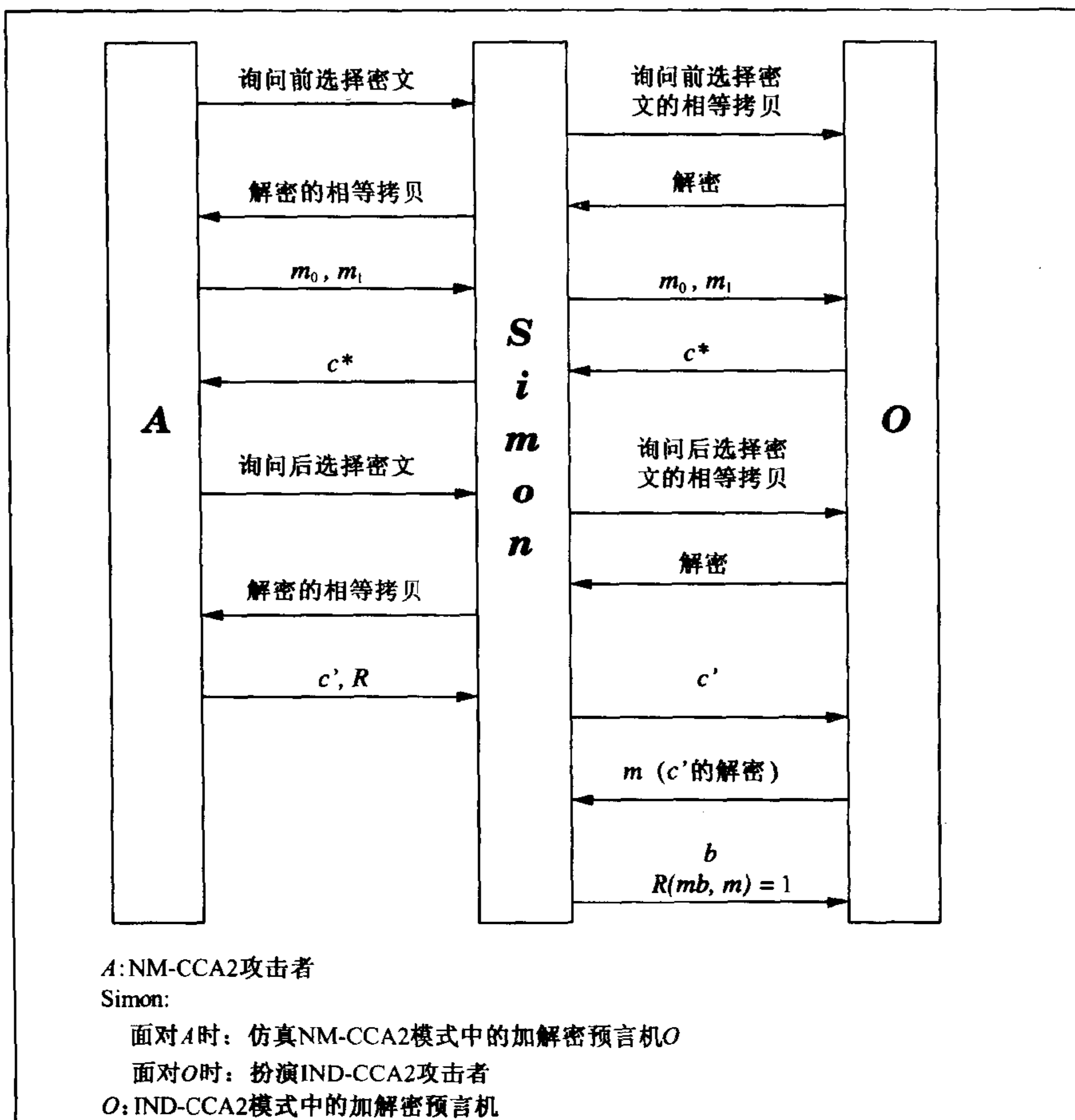


图 14.3 从 IND-CCA2 到 NM-CCA2 的归约

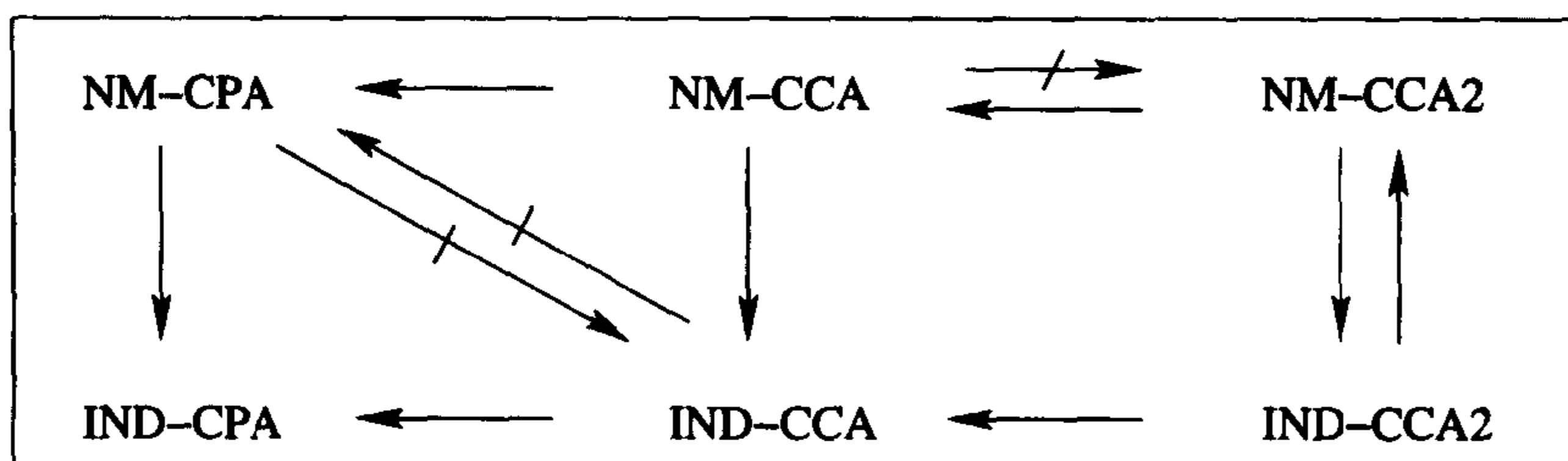


图 14.4 公钥密码体制安全性概念之间的关系

## 14.6 本章小结

本章我们从弱到强,一步步地介绍了公钥密码体制的安全性定义。

从一个使用典型教科书加密算法的协议开始,我们看到了它的弱点以及它的不适于应用性。然后我们介绍了第一步加强的安全性定义:在被动攻击下的语义安全性,或者说不可区分加密。接下来,我们给出了语义安全性的弱点,然后进一步加强安全性定义,直至公钥密码体制最强的安全性定义:在适应性选择密文攻击下的不可区分加密(IND-CCA2),我们认为它是

一个适于应用的安全性定义。最后,我们考虑了公钥体制抗击另一种不同攻击情况的安全性:不可展性,并给出了 IND-CCA2 与不可展性之间的关系。

现如今,IND-CCA2 是公钥密码体制的标准,也是适于应用的安全性概念。所有普通用途的新公钥加密方案必须具备该安全质量。下一章我们将介绍实际的公钥密码体制,它们在 IND-CCA2 攻击模型下是形式化可证明安全的。

## 习题

- 14.1 教科书 RSA 加密能隐藏明文消息 Jacobi 符号的正负吗?
- 14.2 教科书式 RSA(Rabin)加密可以安全地加密工资数据等消息吗? 如果工资数据不在  $\langle g \rangle$  中,那么教科书式 ElGamal 又会如何?
- 14.3 如果在选择明文攻击游戏(协议 14.1)中, $\mathcal{O}$  投掷一个不公平的硬币,正面出现的概率是  $2/3$ ,试推导对应于式(14.2.3)的 Malice 的优势公式。
- 14.4 对于一个公钥加密算法,Malice 有必要做选择明文攻击游戏吗?
- 14.5 什么是语义安全性? 它是针对以下攻击者的吗? i)被动多项式有界攻击者; ii)被动非多项式有界攻击者; iii)主动(多项式有界)攻击者。
- 14.6 语义安全性意味着隐藏明文的所有部分信息。那么为什么它对现实世界的应用还不够强?
- 14.7 如果 Rabin 加密方案(算法 8.2)在午餐攻击(协议 14.3)模型下被攻击,那么攻击者能得到什么?
- 14.8 密码分析训练课程(加解密帮助)是非常有效的,它们为 Malice 提供了攻破所有教科书密码算法的方法。但是我们为什么一般(而且非常慷慨地)都会给予 Malice 这种帮助呢?
- 14.9 什么是 IND-CCA2 安全性? 它针对的是什么攻击?
- 14.10 讨论 IND-CCA2 和 NM-CCA2 安全性等价的重要性。
- 14.11 在凌晨攻击游戏(协议 14.4)中,如果只允许 Malice 提交他用指定加密方案加密得到的密文,试证明该游戏退化为午餐攻击。但是为什么不会进一步退化为 IND-CPA 攻击呢?

## 第 15 章 可证明安全的有效公钥密码体制

### 15.1 引言

在上一章我们看到达到 IND-CCA2(等价地, NM-CCA2)安全的早期公钥密码方案都普遍依赖于非交互式零知识(NIZK)证明技术的应用。这些证明告诉密文的接收者该密文的生成者已经知道相应的明文, 因为被证明的是下述 NP 成员资格<sup>①</sup>的陈述:

“密文  $c$  在语言  $L$  中,  $L$  由公钥  $pk$  下加密的算法  $\mathcal{E}$  定义, 并且  $c$  的生成者拥有一个辅助输入(即 NP 问题的一个证据)可以证明成员资格。”

这里为了证明成员资格, “辅助输入”包括相应的明文, 可能还要加上算法  $\mathcal{E}$  的随机输入(如果需要加密方案是语义安全的, 那么该随机输入是必需的)。虽然接收者可能被要求或者被欺骗提供解密服务, 但是只要对证明的验证输出“Accept”, 他就可以确信即使密文  $c$  的生成者是 Malice(坏人), 将相应的明文返回给 Malice 也只是告诉他一些他已知的信息, 因此不管 Malice 企图用什么方式攻击目标密码体制, 这样做都不会帮助他。

这虽然是一个很好的想法, 但却需要付出昂贵的代价。实现 NIZK 证明的一般方法是证明者(这里是密文的生成者)和验证者(这里是消息的接收者)分享一个互相信任的随机串。这一要求超出了加密方案应该要求的范围。如果我们认为公钥密码学在安全通信中最重要的优势就是去掉了双方分享秘密信息的要求<sup>②</sup>, 那么我们就不能退回到在通信双方中以分享互相信任的信息为代价而建立公钥加密方案的可证明安全性!

事实上, 可证明安全性应该实现的是一种肯定的量度, 确保 Malice 不能过于频繁而且足够快地做坏事。所以只要我们可以确定 Malice 在攻击中成功的概率以及计算代价, 我们就可以建立可证明安全性。在达到可证明安全加密的含义中, 要求 Malice 必须知道密文相应的明文, 这样的保证太过分了, 所以 NIZK 证明是不必要的, 也过火了。事实上, 前面基于 NIZK 证明技术的可证明 IND-CCA2 安全的公钥密码体制在实际应用中都不是太有效。

人们提出了很多实际有效而且可证明安全的公钥加密和数字签名方案。这些方案大都是使用消息完整性检验机制加强通常的教科书公钥算法或数字签名方案而得到的。这里的教科书公钥算法(见 8.14 节)是指直接应用某些陷门单向函数的算法, 如 RSA、Rabin 和 ElGamal 函数。消息完整性检验机制使我们可以确定 Malice 对加强的方案进行攻击的成功概率和计算代价。

使用以这种方式加强的方案的代价是使用基本教科书公钥加密算法所需代价的一个小常数倍。

#### 15.1.1 本章概述

本章我们介绍两个著名的公钥加密方案, 它们对 IND-CCA2 是可证明安全的, 且在实际中

① 在第 18 章我们将研究 NP 成员资格陈述和零知识证明之间的关系。

② 我们看到过甚至不需要双方分享公开信息的安全通信方法, 见 13.3 节。

很有效。它们分别是最优非对称加密填充(OAEP)[25,272,116](见 15.2 节)和 Cramer-Shoup 公钥密码体制[85](见 15.3 节)。接下来我们将概括一类所谓的混合密码体制,它们是公钥和私钥加密算法的组合,它们对 IND-CCA2 也是可证明安全的,而且在实际中很有效(见 15.4 节)。本章最后,我们将回顾关于实用而且可证明安全的公钥密码体制的文献(见 15.5 节)。

## 15.2 最优非对称加密填充

最优非对称加密填充(OAEP)是由 Bellare 和 Rogaway[25]提出的。它是一种随机化的消息填充技术,而且是从消息空间到一个陷门单向置换(OWTP)定义域的一个易于求逆的变换。RSA 和 Rabin 函数是两个最著名的 OWTP<sup>①</sup>。该变换使用两个密码杂凑函数,输入为一条明文消息、一个随机数以及为了使消息可识别而加的作为冗余的一串 0。图 15.1 描述了该变换。使用 RSA-OAEP 方案的具体方法(即该 OWTP 的实例是 RSA 函数)在算法 10.6 中给出。我们应该注意到对于算法 10.6 中介绍的 RSA-OAEP,因为在加密过程中加入了测试步骤,该填充会使得  $s \parallel t$  作为一个整数总是小于 RSA 的模数  $N$ 。

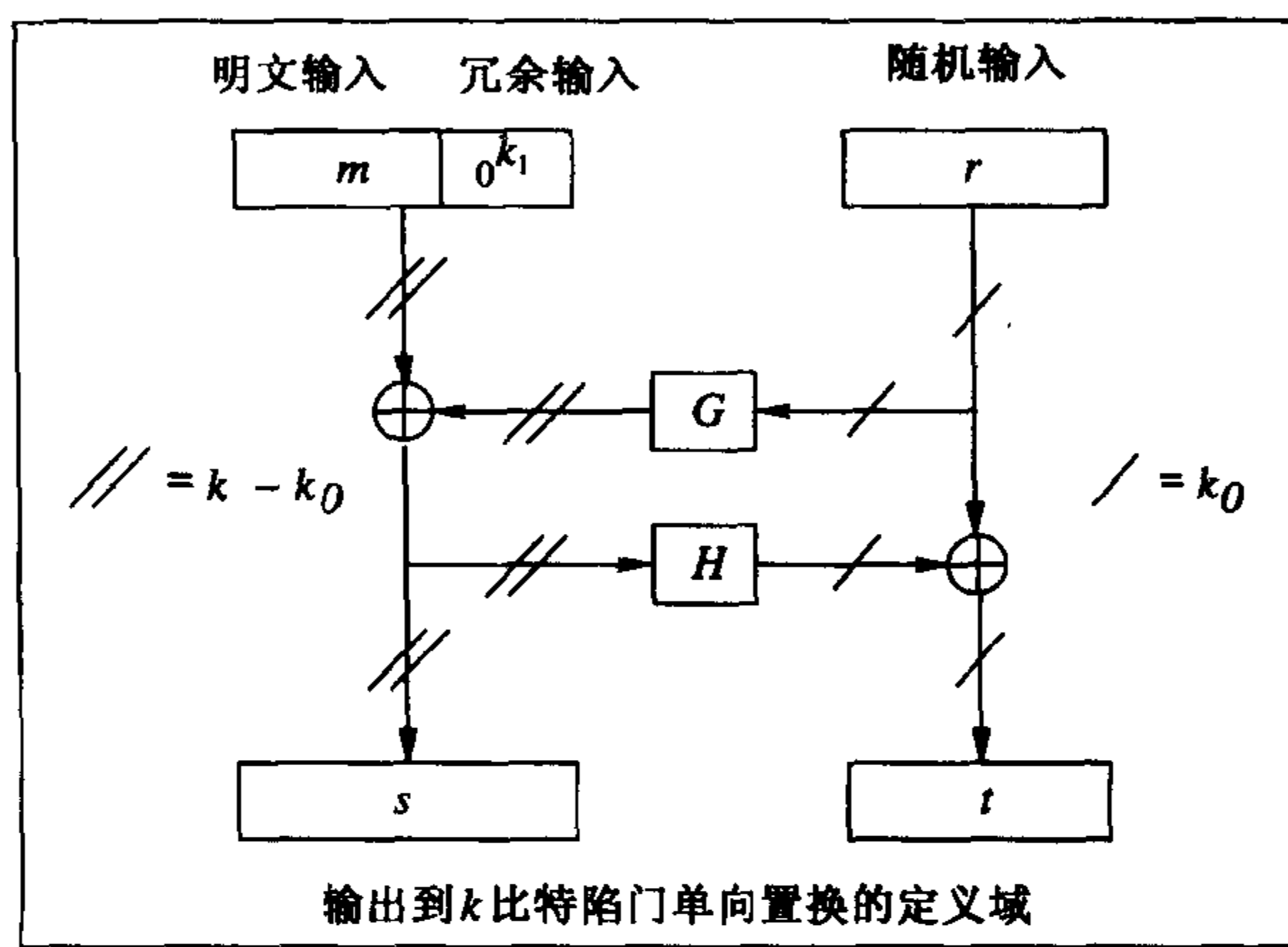


图 15.1 最优非对称加密填充(OAEP)

基于 OAEP 的公钥加密方案可以看做是一个顺序的联合变换：

$$\text{明文} \xrightarrow{\text{OAEP}} \text{OWTP 的定义域} \xrightarrow{\text{OWTP}} \text{密文} \quad (15.2.1)$$

现在我们给出对该联合变换中心思想的三个说明。

**不同代数结构的混合** 正如我们在 8.6 节中讨论过的,教科书公钥算法中的数学函数通常都具有很好的性能和公开的代数特性。这些代数特性来自潜在的代数结构,OWTP 就定义在该结构中(例如,群或域上的公理就提供了很好的代数特性,见第 5 章的定义 5.1 和 5.13)。我们已经介绍过的大量对教科书公钥加密算法的攻击(包括对语义安全方案的攻击,如 GM 体制、算法 14.1、例 14.2 和 14.3 给出对它的攻击)一成不变地展示了 Malice 攻击教科书加密算法的一种通用技术:篡改密文从而以一种可控的方式修改相应的明文,而之所以如此正是因为 OWTP 性能良好的代数特性。

截然不同是,OAEP 变换是把密码学杂凑函数和一个著名对称密码算法结构结合起来构造的。事实上,正如图 15.2(与图 7.2 比较)所示,OAEP 构造可以看做是一个两轮 Feistel 密码,第一轮使用杂凑函数  $G$ ,第二轮使用杂凑函数  $H$ ,代替 Feistel 密码的“s 盒函数”,但是这里的两个“s 盒函数”不是加密钥的,而且两个“半分组”的大小可以不同。

① 关于怎样用一种得当的方法使 Rabin 加密构成 OWTP 以及为什么这样做,可以参考 14.3.6.1 节。



这两种结构,即通用的公钥密码体制所基于的 OWTP 结构和 OAEP 的 Feistel 密码结构,有截然不同的代数特性。粗略地判断,我们可以明显看到前一个结构在大基数空间中具有逐组的代数特性,而后者则具有逐比特(即在基数为 2 的空间中)的代数特性。因此,我们应当很有信心:如果 Malice 想通过篡改密文以一种可控的方式修改相应的明文,组合变换(15.2.1)会给他造成极大的困难。

**明文随机化** 正如我们在第 9 章中学过的,如果一个基本教科书公钥密码函数的输入明文分布随机,那么该函数就为隐藏明文信息提供了一个强的保护,甚至是对单个比特级。像 OAEP 这样的填充方案有一个随机输入值,这为填充结果加上了随机分布,也就是说,它使得 OWTP 的输入分布更随机。所以,为了将这个随机化填充方案与 OWTP 连续组合,我们希望能利用第 9 章中公钥密码原型的强比特安全性。

**数据完整性保护** 我们多次看到教科书密码算法的共同缺陷是抗击主动攻击的能力极差。可逆函数(RSA 解密和 Feistel 网络)以及冗余  $0^k$  使解密以一个检验消息完整性(Malice 提交数据的完整性,见 10.5 节)的机制结束。所以可以防止主动攻击。

如果随机化填充的输出确实在 OWTP 的输入消息空间上具有很好的随机分布,那么以上三点说明就很有意义。

OAEP 加密方案的形式化证明要基于一个非常有效的技术,称为随机预言机模型(ROM)。这种证明假设 OAEP 中使用的杂凑函数的性能完全和随机函数一样,通常称为随机预言机关于随机预言机的行为,请回顾 10.3.1.2 节)。在随机预言机假设下,也就是当在填充中使用的杂凑函数是随机预言机时,填充的输出,即 OWTP 的输入,确实服从均匀分布。因此,我们可以建立一个证明获得类似于第 9 章的结论,这在直觉上似乎很有希望。

精确地讲,对于 OAEP-OWTP 加密方案的基于随机预言机模型的证明,它的目标是构造一个有效变换(称为归约),将对 OAEP-OWTP 加密方案的所谓攻击优势转化为对该方案中所用的 OWTP 求逆的类似(最多相差多项式)优势。例如,当 OWTP 是 RSA 函数时,求逆实际上解决了 RSA 问题或者说攻破了 RSA 假设(见 8.7 节中的定义 8.4、假设 8.3)。因为人们普遍相信不存在有效的算法可以对 OWTP 求逆,所以我们可以认为这个有效的归约导致了一个矛盾。因此,这样构造的证明称为归约为矛盾。

### 15.2.1 安全性证明的随机预言机模型

在 10.3.1.2 节中我们介绍过了随机预言机的概念。随机预言机是一种强大的虚拟函数,它是确定性的、有效的,而且它的输出服从均匀分布。

Bellare 和 Rogaway 在证明 OAEP 加密方案的安全性时利用随机预言机的这些性质[25]。他们的证明模型称为随机预言机模型[23]。

基于 ROM 的安全性证明技术中,不仅使用了随机预言机(即不仅假设它们存在),而且使用了一个特殊的代理——Simon 仿真器,在 14.5.1 节中我们已经遇到过这个概念,它可以以某

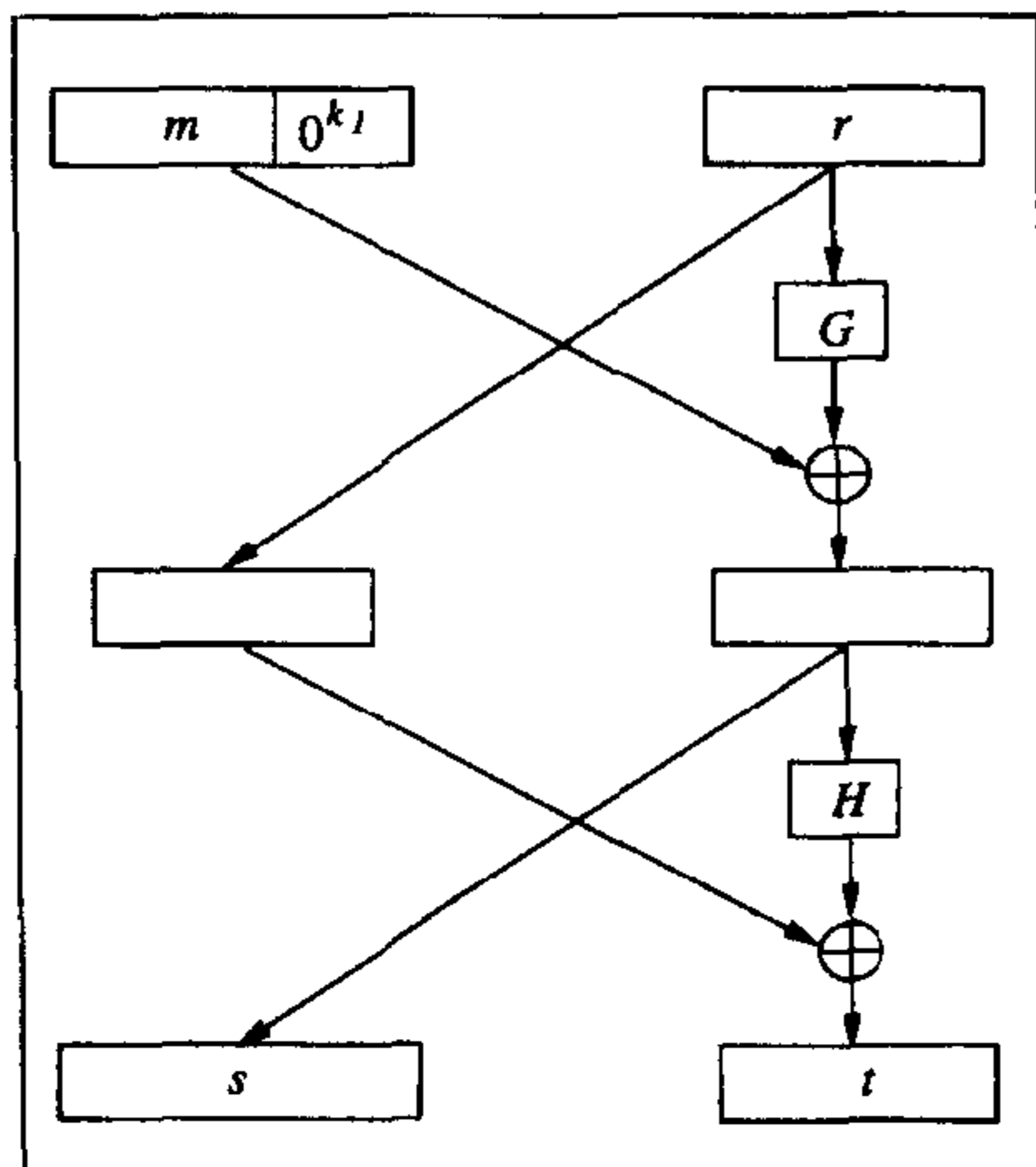


图 15.2 作为两轮 Feistel 密码的 OAEP

种方式仿真每个人(包括 Malice)的随机预言机行为。所以,当某人想对某个值应用一个随机预言机时,譬如对  $a$  应用  $G$ ,他(她)就必须下意识地对 Simon 做一个所谓的随机预言机提问:他(她)提交  $a$  给 Simon,然后从 Simon 那里收到提问结果  $G(a)$ 。Simon 应该总是诚实地遵从任意次序的提问,并且按时返回正确的提问结果。

只要每个人都遵守只提问 Simon 的随机预言机这个规则,Simon 就可以很容易地而且是完全精确地仿真该随机预言机的行为。现在让我们来解释 Simon 怎样仿真随机预言机的行为。

例如,对于预言机  $G$ ,Simon 要维护一个  $G$  表,它包含所有的  $(a, G(a))$  对, $a$  是在  $G$  的整个历史中被提问过的值。仿真工作很平凡:对每一个提问  $a$ ,Simon 检查  $a$  是否已经在表中,如果是,他就将  $G(a)$  作为提问结果(这就是为什么是确定性的)返回;否则,他在  $G$  的值域范围内均匀随机地选取一个新值作为  $G(a)$ ,将这个新值作为提问结果(这就是为什么是均匀的)返回,并将新的  $(a, G(a))$  对记录在表中。Simon 可以将他的表以第一个元素排序。没有必要使用排序算法,因为开始时每个表都初始化为空,然后随着提问的增加而增加。对于每一个提问,在  $N$  个排过序的元素中查找可以在  $\log N$  时间内完成(见算法 4.4),也就是在关于元素大小的多项式时间内(这就是为什么是有效的)完成。

现在,我们构造性地得到了引理 15.1。

**引理 15.1** 随机预言机可以在 PPT 内完全仿真。

对于使用随机预言机的公钥加密方案(如 OWTP-OAEP),这种仿真随机预言机的方法使得 Simon 可以在明文和密文之间构造一个一一映射关系[例如式(15.2.1)中的 OWTP-OAEP,将左边的映射到右边]。现在,如果一个攻击者,如 Malice,使用 OWTP  $f$  构造了一个有效的选择密文,那么只要 Malice 在构造  $c$  时使用过 Simon 的随机预言服务(他是被迫使用的),Simon 就可以“解密” $c$ ,即使他没有对  $f$  求逆的陷门信息。这仅仅是因为 Simon 在他仿真随机预言机的表中有这对明密文。事实上,只要密文是合法的,该明文就一定在他的某个表中。

因此除了仿真随机预言机,Simon 还可以仿真解密预言机,即协议 14.3 或协议 14.4 中的  $\mathcal{O}$ 。这也是我们称特殊的代理为 Simon 仿真器<sup>①</sup>的另一个原因。仿真的“解密”能力使得 Simon 可以在 IND-ATK 游戏(ATK 表示 CPA、CCA、CCA2 中的任何一个)中,为 Malice 提供一个正确的“密码分析训练课程”。

如果提供精确的“训练课程”(即仿真的“训练课程”是准确的),那么作为一个成功的攻击者,Malice 就必定以一个不可忽略的优势,在足够短的时间内(PPT)攻破加密方案而结束(即在 IND-ATK 中,他将两个选择明文之一与询问密文联系起来而结束)。然后,拥有随机预言机的 Simon 也会以针对这一条询问密文成功求得密码函数的逆而结束:该明文-密文询问对会出现在他仿真随机预言机的其中一个列表中。对于 OWTP-OAEP,我们在 15.2.3.1 节中会给出该“技巧”的详细情况。下一章我们还会看到,基于 ROM 对某些数字签名方案的安全性证明中,是如何使用这一“技巧”的。

上述内容确实构成了一个有效的论据,但只是对存在随机预言机的世界而言!

尽管如此,因为密码杂凑函数能很好地仿真随机预言机的行为,所以这个论据为用 OAEP

<sup>①</sup> 读者一定不要混淆“解密预言机”和“随机预言机”,它们是完全不同的。前者可以是真实的,例如,一个天真的用户被 Malice 欺骗提供了解密服务,而后者只是一个虚拟的函数。

加强的加密方案在现实世界中确实安全提供了一种令人信服的启发式见解。虽然我们知道,密码杂凑函数在 PPT 无法识别的方式下可以仿真随机预言机的行为,这还只是一个未被证明的假设。但这一假设已被人们广泛地接受,并在实际中使用。毕竟每个通用的公钥密码体制使用的著名 OWTP 都是未被证明却被广泛接受的假设。

Goldreich 认为([66]的“6.2 节 Oded 的结论”中)基于 ROM 证明安全性的技术是一个有用的试验台:在该试验台上性能不好的密码体制(即不能通过这个合乎情理的检查)应该被丢弃。人们广泛认为设计一个密码学方案,使其在 ROM 中可证明安全是一个很好的工程准则。

在真实世界中,如果密码学方案或协议中使用的杂凑函数(或伪随机函数)没有“明显”的缺点,那么使用它们的理想化模型对这些方案或协议证明安全性,尤其是当证明的目的只需要达到尚未证明的 PPT 时间不可区分性假设时,我们可以认为这样的证明是有效的。这样的安全性证明称为基于 ROM 的安全性证明。

现在让我们描述对一个 OWTP-OAEP 加密方案的基于 ROM 的安全性证明。在下一章,我们还会看到对某些数字签名方案的基于 ROM 的安全性证明。

### 15.2.2 RSA-OAEP

对于 RSA-OAEP,OWTP 就是 RSA 加密函数。注意本章的其余部分中,所有 RSA-OAEP 实例都适用于 Rabin-OAEP,这里 OWTP 是 Rabin 加密函数的置换实现(关于怎样将 Rabin 加密函数实现成一个 OWTP,见 14.3.6.1 节)。

因为 RSA-OAEP 加密方案在使用 RSA 函数(算法 10.6)之前涉及到两个杂凑函数的计算,而杂凑函数可以有效地计算,所以该方案非常有效,几乎与教科书 RSA 一样有效。对于消息恢复该方案也有很高的带宽。如果我们考虑使用标准长度为 2048 比特的 RSA 模数(对 RSA-OAEP 采用 2048 比特为标准长度的原因将在 15.2.5 节中解释),而且  $k_0 = k_1 = 160$ (使得  $2^{-k_0}$  和  $2^{-k_1}$  小到可以忽略),那么明文消息的长度可以是  $|M| = |N| - k_0 - k_1 = 2048 - 320 = 1728$ ,也就是在 RSA-OAEP 中要加密的明文消息可以达到模数长度的 84%。

实践者们广泛地认识到了这些在实际中非常重要的特性,所以该方案被国际工业标准组织接受为 RSA 的加密标准(PKCS #1, IEEE P1363)。它也被著名的 Internet 电子商务协议 SET 选中使用[261]。

所以,RSA-OAEP 是一个非常成功的公钥加密方案。但是,对于它的可证明安全性来说,失败是成功之母。

如果读者只想知道怎样用 RSA 加密才能获得适于应用的安全性,那么算法 10.6 中的 RSA-OAEP 方案提供了足够关于“知其然”的信息,读者可以直接学习 15.3 节。从这里一直到 15.3 节之前的内容都是关于“知其所以然”的材料:回答为什么 RSA-OAEP 方案具有适于应用的安全性。我们将努力以直观的方式给出答案,并讨论与安全性证明有关的一些重要问题。

### 15.2.3 RSA-OAEP 证明中的曲折

对于  $f$ -OAEP,最初基于 ROM 的证明[25]试图将对  $f$ -OAEP 方案的 IND-CCA2 攻击与在不用  $f$  的陷门信息的情况下对 OWTP  $f$  求逆的问题联系起来。最近 Shoup 巧妙地发现并揭示了该证明中的一个缺点[272]。而且,他还指出对于  $f$  是 OWTP 的一般情况,该  $f$ -OAEP 对 IND-CCA2 安全性基于 ROM 的证明可能是不存在的。所幸的是,让我们失去一个非常成功的公钥

加密算法标准的危险很快就成为过去。Fujisaki 等人[116]有了进一步的发现,他们找到了当  $f$  是 RSA 函数时挽救 OAEP 的方法。

现在让我们回顾一下这个戏剧性的过程。我们从学习 Bellare 和 Rogaway 最初的安全性证明开始,然后描述 Shoup 在该证明中发现的一个问题,最后介绍 Fujisaki 等人的挽救工作(我们也将看到 Shoup 对同样的挽救工作也提出了一个特例)。

### 15.2.3.1 基于随机预言机模型的归约

假定一个 PPT 算法的攻击者  $A$  可以以一个不可忽略的优势在 IND-CCA2 模式下攻破  $f$ -OAEP 方案。我们构造一个算法可以使特殊的代理——Simon 仿真器——利用 IND-CCA2 攻击者  $A$  对 OWTP  $f$  求逆,其优势也是不可忽略的。这个算法一定是有效的(即 PPT 的)。这样,Simon 就有效地将对  $f$  求逆的任务“归约”到了  $A$  攻破  $f$ -OAEP 的能力。因此 Simon 使用的算法就称为多项式时间归约。因为  $A$  和 Simon 的归约都是多项式时间的,所以组合  $A$  和 Simon 的归约对  $f$  求逆的算法也是多项式时间的。人们相信对  $f$  求逆不可能在 PPT 内完成,这与对  $f$ -OAEP 所谓的 IND-CCA2 攻击者的存在性相矛盾(但是,我们要留意即将在 15.2.5 节中讨论的情况)。这种形式的安全性证明,除了称为“归约为矛盾”以外,又称为归约证明。

现在我们将描述这个归约。

假设给定 Simon 一个 OWTP  $f$ (的描述)和一个在  $f$  值域中均匀分布的  $c^*$ 。Simon 想通过使用 IND-CCA2 攻击者  $A$  得到  $f^{-1}(c^*)$ 。我们必须注意  $c^*$  的随机性非常重要:如果  $c^*$  不是随机的,那么 Simon 的结果就不是一个有用的算法。

#### 归约算法的顶层描述

图 15.3 将帮助我们形像化地了解现在要描述的归约。该图表明 Simon 控制了  $A$  与外界的全部通信渠道,这样  $A$  就只能跟 Simon 通信。

- Simon 以发送该  $f$ -OAEP 加密算法(的描述)给  $A$  而开始。
- Simon 与  $A$  进行一个 IND-CCA2 攻击游戏(即运行协议 14.4)。在该游戏中,Simon 扮演解密预言机  $O$ ,就像他拥有一个有效的解密盒一样。这是通过仿真来实现的。我们会看到,在 ROM 中,Simon 确实可以这样做,而且  $A$  不会发现任何错误。
- Simon 还给  $A$  提供 OAEP(见图 15.1)中使用的随机预言机  $G$  和  $H$  的仿真服务。所以,正如我们在 15.2.1 节中规定的,无论何时  $A$  想使用  $G$  和/或  $H$ (即当  $A$  在攻击游戏过程中想以正确的方式准备一个选择密文时),它实际上都要提问 Simon,然后从 Simon 那里得到相应的提问结果。

至关重要的是,Simon 必须提供准确的仿真,这样才能使  $A$  在与外界通信时不会感到出了错。只有在精确的仿真下,Simon 才能正确地教  $A$  使其完全发挥它的攻击才能。Simon 和  $A$  之间的 IND-CCA2 攻击游戏如下:

- i) 在  $A$  的“寻找阶段”,Simon 从  $A$  收到一些冷漠选择密文要求解密(即午餐攻击中的密文)。 $A$  可以以它希望的任何方式构造这些密文;但如果它确实想正确地构造这些密文,例如通过应用随机预言机,那么它的提问都必须发送给 Simon(正如图 15.3 所示,Simon 控制了  $A$  与外界的全部通信)。Simon 仿真这些随机预言机的方法我们很快就会描述。

- ii) 因为 Simon 从  $A$  收到一些选择密文要求解密, 所以他必须通过仿真解密盒(预言机  $\mathcal{O}$ ) 回答  $A$ 。Simon 仿真  $\mathcal{O}$  的方法我们也会很快描述。
- iii)  $A$  提交给 Simon 一对选择明文  $m_0, m_1$  以结束它的“寻找阶段”。Simon 一收到它们就抛一个公平的硬币  $b \in_v \{0, 1\}$ , 将“询问密文”  $c^*$  作为  $m_b$  的仿真  $f$ -OAEP 加密发送给  $A$ 。这里, Simon 假装  $c^*$  是  $m_b$  的加密。
- iv) 现在  $A$  处于“猜测阶段”, 所以它可以进一步提交适应性选择密文获得它的“扩展密码分析训练课程”。Simon 提供与(ii)中一样的服务。在  $A$  用正确构造的适应性选择密文提问随机预言机的情况下, Simon 提供与(i)中一样的服务。
- 最终,  $A$  应输出它对比特  $b$  经过训练的猜测。这就结束了攻击游戏。

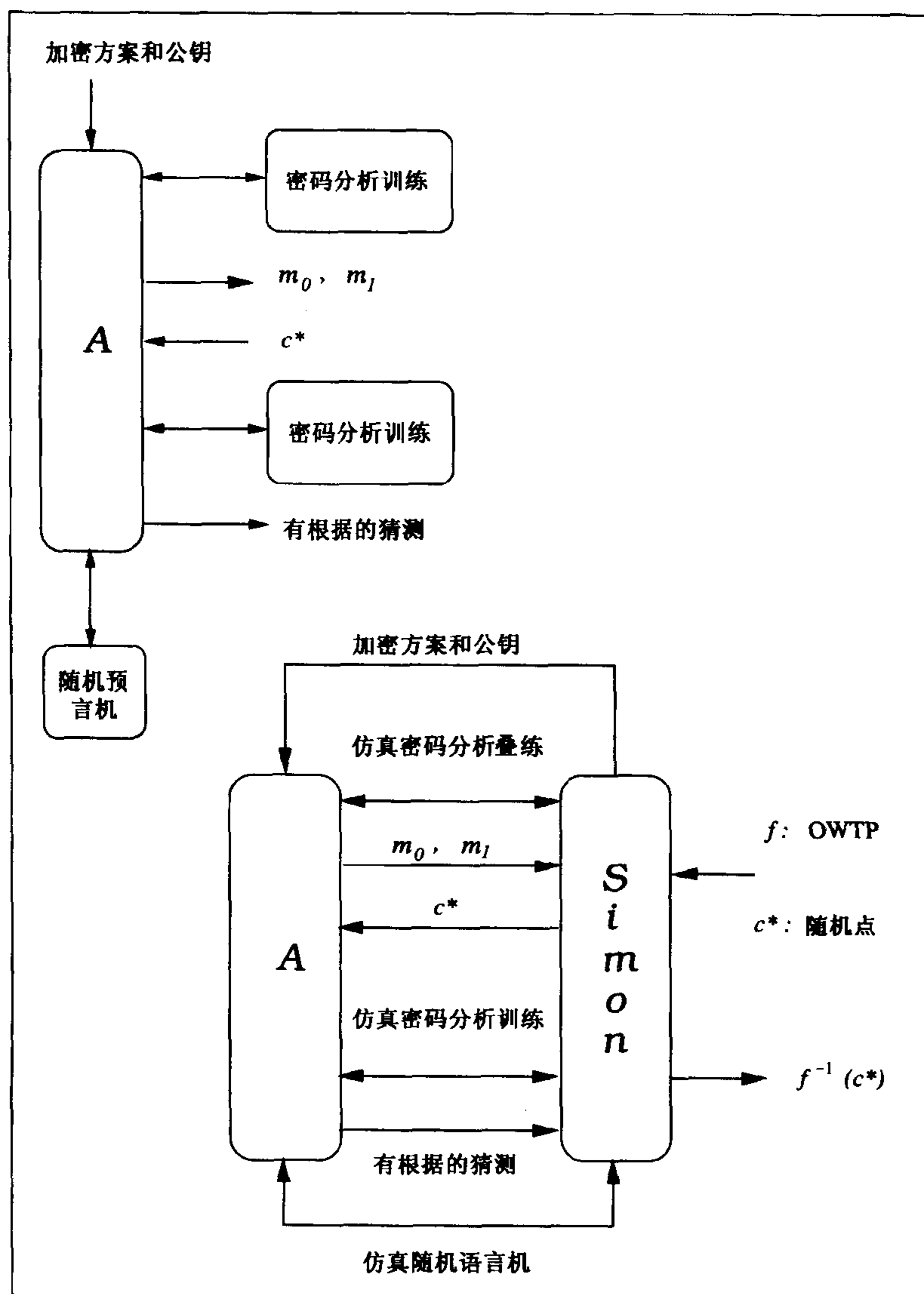


图 15.3 从陷门单向函数  $f$  的求逆到对  $f$ -OAEP 方案的一个攻击的归约



我们已经多次指出并强调 $\mathcal{A}$ 不能提交“询问密文” $c^*$ 要求解密。如果提交 $c^*$ , Simon 就不可能提供仿真解密,因为这正是 Simon 需要 $\mathcal{A}$ 帮助解密的密文。

**随机预言机的仿真。** Simon 要仿真 OAEP 变换中用到的两个随机预言机  $G$  和  $H$ 。在仿真中, Simon 维护两个表,分别称为他的  $G$  表和  $H$  表,初始时都设定为空:

**$G$  预言机** 假设 $\mathcal{A}$ 做了一个  $G$  提问  $g$ 。Simon 首先在他的  $G$  表中寻找  $g$ 。如果  $g$  在表中, Simon 给 $\mathcal{A}$ 提供;否则,  $g$  是新值, Simon 就均匀随机地选择一个长度为  $k_0$  的串  $G(g)$ , 将  $G(g)$  提供给 $\mathcal{A}$ , 并将新的对  $(g, G(g))$  添加到  $G$  表中。

如果提问发生在 $\mathcal{A}$ 的“猜测阶段”, Simon 就会试图在  $c^*$  点对  $f$  求逆。他需要做的是: 对每个  $(g, G(g)) \in G$  表以及  $(h, H(h)) \in H$  表, Simon 构造  $w = h \parallel (g \oplus H(h))$  并检查是否有  $c^* = f(w)$ 。如果对这样构造的某个串成立, 那么  $f^{-1}(c^*)$  就找到了。

**$H$  预言机** 假设 $\mathcal{A}$ 做了一个  $H$  提问  $h$ 。Simon 首先在他的  $H$  表中寻找  $h$ 。如果  $h$  在表中, Simon 给 $\mathcal{A}$ 提供  $H(h)$ ; 否则,  $h$  是新值, Simon 就均匀随机地选择一个长度为  $k - k_0$  的串  $H(h)$ , 将  $H(h)$  提供给 $\mathcal{A}$ , 并将新的对  $(h, H(h))$  添加到  $H$  表中。

如果提问发生在 $\mathcal{A}$ 的“猜测阶段”, Simon 就会和  $G$  预言机的情况一样试图在  $c^*$  点对  $f$  求逆。

**解密预言机的仿真。** Simon 要仿真解密盒(预言机 $\mathcal{O}$ )。他的仿真步骤是: 一旦从 $\mathcal{A}$ 收到密文  $c$  要求解密, 对每一对提问回答  $(g, G(g)) \in G$  表和  $(h, H(h)) \in H$  表, Simon 就计算

$$\begin{aligned} w &= h \parallel (g \oplus H(h)) \\ v &= G(g) \oplus h \end{aligned}$$

然后检查

$$c \stackrel{?}{=} f(w)$$

以及

$$v \text{ 是否有连续 } k_1 \text{ 个 } 0?$$

如果两个检查步骤都回答“YES”, Simon 就返回  $v$  的  $n = k - k_0 - k_1$  个高位比特给 $\mathcal{A}$ 。否则 Simon 返回“REJECT”。

因为 $\mathcal{A}$ 是多项式有界的, 所以它提问随机预言机和解密机的次数也是多项式有界的。因此, Simon 可以在多项式时间内运行这个仿真游戏。

### 15.2.3.2 仿真的精确性

正如我们已经提到的, 为了使 $\mathcal{A}$ 能正确工作, 仿真的精确性是至关重要的(就是一切)!

首先, 正如我们在引理 15.1 中所确立的, 可以完全仿真这两个随机预言机。

现在, 让我们检验一下 Simon 对解密盒仿真的精确性。

假设给定 Simon 一条选择密文  $c$  (或者是询问前的, 或者是询问后的, 即或者是冷漠选择的, 或者是适应性选择的)。Simon 对解密盒的仿真实际上很精确。如果  $c$  是一条有效密文, 令

$$s \parallel t = f^{-1}(c) \quad (15.2.2)$$

$$r = t \oplus H(s) \quad (15.2.3)$$



$$m \parallel 0^{k_1} = s \oplus G(r) \quad (15.2.4)$$

是由  $c$  定义的值。如果  $c$  被发送给  $\mathcal{O}$ , 以下我们所说的“正确值”是指由真正的解密预言机  $\mathcal{O}$  得到的值。

如果由  $c$  在式(15.2.2)中定义的正确值  $s$  还没有向随机预言机  $H$  提问, 那么正确的  $H(s)$  就不出现了。所以在每次  $G$  提问中, 式(15.2.3)中定义的  $r$  正确的概率只有  $2^{-k_0}$ 。与向  $H$  提问的  $s$  的正确值丢失一样, 向  $G$  提问的  $r$  的正确值也不出现(除去  $2^{-k_0}$  的概率)。因此, 如式(15.2.4), 值  $s \oplus G(r)$  只能以  $2^{-k_1}$  的概率有连续  $k_1$  个 0, 因为这要求  $s$  和  $G(r)$  的  $k_1$  个低位比特逐比特相等, 但是前者已经丢失, 而后者是均匀随机的。注意, 在该分析中, 我们也已经考虑了正确的  $r$  没有向  $G$  提问的情况: 拒绝是正确的, 除了  $2^{-k_1}$  的错误概率。

总之, 关于对选择密文  $c$  的仿真解密, 我们可以得出下列结论:

- 如果分别向随机预言机提问过  $r$  和  $s$ , 那么仿真解密可以正确地构造  $f^{-1}(c)$ , 因此可以像通常一样解密  $c$ 。
- 如果没有向随机预言机分别提问过  $r$  和/或  $s$ , 那么仿真解密返回 REJECT 是正确的, 除非有  $2^{-k_0} + 2^{-k_1}$  的概率犯错误。

注意, 在没有分别向随机预言机提问过  $r$  和/或  $s$  的情况下, 错误概率界在统计意义下成立: 即只要  $\mathcal{A}$  没有做必要的随机预言机提问, 这个概率界就成立, 不管  $\mathcal{A}$  有多么强大。

现在我们可以确定目前的论据已经说明 OAEP 对 IND-CCA (即午餐攻击或者说冷漠选择密文攻击) 是可证明安全的。这是因为在 IND-CCA 攻击游戏中, 解密盒只在“寻找阶段”工作, 我们已经证明在该阶段中, 仿真解密可以准确地工作, 除了一个微小的错误概率。

我们必须明确强调上述证据只是基于  $f$  的单向性。

### 15.2.3.3 不完全性

Shoup 发现最初的 OAEP 安全性证明中(对 IND-CCA2)有一个缺陷[272]。在我们解释该缺陷之前, 要先明确 OAEP 的构造没有缺陷。只是 15.2.3.2 节中描述的形式化证明没有完全完成而到达 IND-CCA2 安全性。对这个不完全性, 我们简短陈述如下:

只要由询问密文  $c^*$  在式(15.2.5)中定义的  $s^*$  没有向随机预言机  $H$  提问, Simon 的仿真解密就是统计上精确的, 但是一旦  $s^*$  被提问, 该统计精确性就崩溃了。然而, 在 15.2.3.2 节的安全性证明中, 没有考虑  $s^*$  被提问的概率。

为了解释这个不完全性, 让我们考虑由询问密文  $c^*$  定义各个值。令

$$s^* \parallel t^* = f^{-1}(c^*) \quad (15.2.5)$$

$$r^* = t^* \oplus H(s^*) \quad (15.2.6)$$

$$m_b \parallel 0^{k_1} = s^* \oplus G(r^*) \quad (15.2.7)$$

$(s^*, r^*, m_b)$  这三个值是由询问密文  $c^*$  定义的, 这里  $b$  是 Simon 硬币投掷的结果。

现在假设向随机预言机  $H$  提问  $s^*$ 。当然, 在统计上这在  $\mathcal{A}$  的“寻找阶段”是远远不可能发生的, 因为这个阶段它还没有询问密文  $c^*$ 。也正是这个原因, 我们在 15.2.3.2 节结束时得到以下结论: 那里得到的论据确实提供了  $f$ -OAEP 在 IND-CCA 模式下安全的有效证明。但是, 当给定  $\mathcal{A}$  询问密文  $c^*$  后, 它在“猜测阶段”是有可能提问  $s^*$  的。

那么  $A$  在“猜测阶段”提问的概率有多大呢？这个问题我们还不明确。我们明确知道的是：给定一个所谓强大的  $A$ ，我们不能否认它在“猜测阶段”会提问  $s^*$  的概率。否则，我们为什么还要在一开始就假设  $A$  可以猜测比特  $b$  呢？（尽管如此，在  $A$  正确回答的条件下， $s^*$  被提问的条件概率界是可以估计的：结果是不可忽略的。我们将会在第 15.2.3.4 节中给出一个估计。）

只要  $A$  可以提问  $s^*$ ，它就可以在仿真攻击游戏中找到不符之处。我们有一个很简单的方法设想这一不符。对于式 (15.2.6) 中固定的  $r^*$ ， $A$  可能会进一步提问  $r^*$ 。返回的均匀分布的  $G(r^*)$  只能以很小的概率使  $s^* \oplus G(r^*)$  与两条选择明文中的一条相同。所以这时  $A$  会大喊：“不要再骗我了！”它这样喊的原因是它认出“询问密文” $c^*$  与它选择的两条明文毫无关系。

当然，这种发现不符之处的“简单”方法会使  $A$  付出很大的代价：它已经将  $s^*$  和  $t^* = r^* \oplus H(s^*)$  暴露给了 Simon，所以也早就帮助 Simon 在点  $c^* = f(s^* \parallel t^*)$  处对  $f$  求了逆！

Shoup 很好地利用了这一问题。他发现对某个作为 OWTP 的  $f$ ，给定  $c^*$  后， $A$  提问  $s^*$  的能力使它不用向随机预言机  $G$  提问  $r^*$  就足以构造一个有效密文（事实上，在攻击游戏的整个历史中都可以不向  $G$  提问）。而且，因为这样构造的有效密文是另一个有效密文的可展结果，所以这个  $f$ -OAEP 方案不是 NM-CCA2 安全的，由定理 14.4（见 14.5.4.2 节），该方案也不是 IND-CCA2 安全的。但是，15.2.3.1 节中描述的归约技术不会帮助 Simon 求逆  $f$ ，因为 Simon 的  $G$  表甚至可以是空的（即  $A$  没有向随机预言机  $G$  提问任何值）！

Shoup 构造了一个  $k$  比特的 OWTP  $f$  作为一个反例。他假设该置换是“异或-可展的”：给定  $f(w_1), w_2$ ，可以以一个不可忽略的优势构造  $f(w_1 \oplus w_2)$ 。注意这并不是不合理的假设。在 15.2.3.1 节描述的对  $f$ -OAEP 的安全性证明中，我们只要求  $f$  是单向的，并没有要求它是不可展的。归根结底，正如我们在前面的章节中看到的，通常的教科书公钥加密算法采用的 OWTP 一般都是可展的，而这种一般可展性恰恰就是我们用 OAEP 技术提高教科书公钥方案安全性的原因。

为了更清楚地说明，我们假设  $f$  根本不能隐藏  $k - k_0$  个高位比特。这样  $f$  就可以写成：

$$f(s \parallel t) = s \parallel f_0(t)$$

其中  $f_0$  是一个“异或-可展的” $k - k_0$  比特 OWTP，即给定  $f_0(t_1), t_2$ ，可以以一个不可忽略的优势构造  $f_0(t_1 \oplus t_2)$ 。这个  $f$  仍然是一个 OWTP，其单向性由安全参数  $k_0$  度量。

现在考虑用这个  $f$  实现的  $f$ -OAEP。记住  $c^*$  是询问密文， $s^*, t^*, r^*$  和  $(m_b \parallel 0^{k_1})$  是在该  $f$ -OAEP 方案下与  $c^*$  对应的值。

因为  $A$  是一个黑盒，所以我们可以任意地描述怎样通过修改一条有效密文构造另一条有效密文。收到询问密文  $c^*$  后， $A$  分解  $c^*$  为  $c^* = s^* \parallel f_0(t^*)$ ，然后选择一条任意的非 0 消息  $\Delta \in \{0, 1\}^{k - k_0 - k_1}$ ，计算

$$s = s^* \oplus (\Delta \parallel 0^{k_1}), v = f_0(t^* \oplus H(s^*) \oplus H(s)), c = s \parallel v$$

显然，为了从询问密文  $c^*$  构造新的密文  $c$ ， $A$  只需向  $H$  提问  $s^*$  和  $s$ 。

我们现在证明只要  $c^*$  是  $m_b$  的一个有效  $f$ -OAEP 加密，那么  $c$  就是  $m_b$  的有效  $f$ -OAEP 加密。由  $t = t^* \oplus H(s^*) \oplus H(s)$ ，我们有

$$r = H(s) \oplus t = t^* \oplus H(s^*) = r^* \quad (15.2.8)$$

显然，即使  $A$  没有向  $G$  提问过  $r = r^*$ （它甚至可能不知道  $r^*$ ，因为它可能不知道  $t^*$ ），式 (15.2.8) 也成立。

如果这个游戏在 $\mathcal{A}$ 和真正的解密预言机 $\mathcal{O}$ 之间进行,那么 $\mathcal{O}$ 能通过下式计算正确地恢复 $r$ :

$$\begin{aligned} f^{-1}(c) &= s \parallel t \\ r &= H(s) \oplus t \end{aligned}$$

但是注意到式(15.2.8), $\mathcal{O}$ 确实得到了 $r^*$ 。 $\mathcal{O}$ 进一步应用杂凑函数 $G$ ,计算

$$G(r) \oplus s = G(r^*) \oplus s^* \oplus (\Delta \parallel 0^{k_1}) = (m_b \oplus \Delta) \parallel 0^{k_1}$$

一旦看到后面连续的 $k_1$ 个0, $\mathcal{O}$ 就会返回 $m_b \oplus \Delta$ 作为 $c$ 的正确解密结果。从返回的明文 $m_b \oplus \Delta$ , $\mathcal{A}$ 可以很容易地得到 $m_b$ ,从而在 IND-CCA2 模式下攻破该 $f$ -OAEP。

但是,因为这个游戏是在 $\mathcal{A}$ 和 Simon 之间的归约中进行的,而 Simon 的 $G$ 表是空的,所以他会立刻返回 REJECT。这样, $\mathcal{A}$ 无疑就会大喊:

“别再骗我了!”

#### 15.2.3.4 $\mathcal{A}$ 在“猜测阶段”提问过 $s^*$ 的概率

关于 Bellare-Rogaway 最初证明的不完全性,我们遗漏了一个很小的细节: $\mathcal{A}$ 可以正确回答询问比特的条件下,它在“猜测阶段”提问过 $s^*$ 的条件概率。因为这一部分涉及复杂的概率估计,读者可以跳过而不会影响理解 RSA-OAEP 的安全性证明是如何工作的(事实上,该概率估计是非常基础的,用到的所有规则,我们都将指出它们在第3章中的出处)。

首先,我们假设 $\mathcal{A}$ 在得到足够的选择密文训练之后,他正确猜测询问比特 $b$ 的优势是 Adv。

在训练过程中,Simon 可能会错误地拒绝解密一条有效的提问密文。这是一件糟糕的事,因为这表明 $\mathcal{A}$ 的训练课程质量很低。用 DBad 表示这一事件。在 15.2.3.2 节中我们对 Simon 仿真解密过程的检验得出:该仿真解密是精确的或者说是高质量的,不正确的概率只有 $2^{-k_0} + 2^{-k_1}$ 。所以我们有

$$\text{Prob}[\text{DBad}] \approx 2^{-k_0} + 2^{-k_1} \quad (15.2.9)$$

进一步,用 AskG(或 AskH)表示事件 $r^*$ 在 $G$ 表中(或 $s^*$ 在 $H$ 表中)。这两个事件我们都不希望发生,因为它们向 $\mathcal{A}$ 泄漏了信息,使其可以发现询问密文 $c^*$ 实际上与“选择明文” $m_0, m_1$ 没有关系。因此,我们也把它们称为糟糕事件。定义事件 Bad 为

$$\text{Bad} = \text{AskG} \cup \text{AskH} \cup \text{DBad}$$

现在,令 $\mathcal{A}$  wins 表示 $\mathcal{A}$ 正确猜测询问比特 $b$ 这一事件。显然在事件 Bad 没有发生的情况下,由于随机预言机可以具有均匀随机的值,所以询问比特 $b$ 与询问密文 $c^*$ 是独立的。这样我们就有

$$\text{Prob}[\mathcal{A} \text{ wins} | \overline{\text{Bad}}] = \frac{1}{2} \quad (15.2.10)$$

应用条件概率(3.4.1 节的定义 3.3)我们可以将式(15.2.10)进一步表示为

$$\text{Prob}[\mathcal{A} \text{ wins} | \overline{\text{Bad}}] = \frac{\text{Prob}[\mathcal{A} \text{ wins} \cap \overline{\text{Bad}}]}{\text{Prob}[\overline{\text{Bad}}]} = \frac{1}{2}$$

或

$$\text{Prob}[\mathcal{A} \text{ wins} \cap \overline{\text{Bad}}] = \frac{1}{2} \cdot (1 - \text{Prob}[\text{Bad}]) \quad (15.2.11)$$

我们要注意在事件  $\overline{\text{Bad}}$  中(即事件 Bad 不发生),仿真的随机预言机以及仿真的解密盒工作良好,并且与相应的真函数相等。所以  $\mathcal{A}$  的攻击优势将完全发挥,从而有

$$\text{Prob}[\mathcal{A} \text{ wins}] = \frac{1}{2} + \text{Adv} \quad (15.2.12)$$

另一方面(见 3.4.3 节中的定理 3.1:完全概率公式),

$$\text{Prob}[\mathcal{A} \text{ wins}] = \text{Prob}[\mathcal{A} \text{ wins} \cap \overline{\text{Bad}}] + \text{Prob}[\mathcal{A} \text{ wins} \cap \text{Bad}] \quad (15.2.13)$$

联立式(15.2.12)和式(15.2.13),有

$$\text{Prob}[\mathcal{A} \text{ wins} \cap \overline{\text{Bad}}] + \text{Prob}[\mathcal{A} \text{ wins} \cap \text{Bad}] = \frac{1}{2} + \text{Adv}$$

或

$$\text{Prob}[\mathcal{A} \text{ wins} \cap \overline{\text{Bad}}] + \text{Prob}[\text{Bad}] \geq \frac{1}{2} + \text{Adv} \quad (15.2.14)$$

再由式(15.2.11),不等式(15.2.14)就是

$$\frac{1}{2} \cdot (1 - \text{Prob}[\text{Bad}]) + \text{Prob}[\text{Bad}] \geq \frac{1}{2} + \text{Adv}$$

也就是

$$\text{Prob}[\text{Bad}] \geq \text{Adv} \quad (15.2.15)$$

因为  $\text{Bad} = \text{AskG} \cup \text{AskH} \cup \text{DBad}$ , 所以有

$$\text{Prob}[\text{Bad}] \leq \text{Prob}[\text{AskG} \cup \text{AskH}] + \text{Prob}[\text{DBad}] \quad (15.2.16)$$

$$= \text{Prob}[\text{AskH}] + \text{Prob}[\text{AskG} \cap \overline{\text{AskH}}] + \text{Prob}[\text{DBad}] \quad (15.2.17)$$

$$\leq \text{Prob}[\text{AskH}] + \text{Prob}[\text{AskG} | \overline{\text{AskH}}] + \text{Prob}[\text{DBad}] \quad (15.2.18)$$

式(15.2.16)是由于概率加法准则 1, 式(15.2.17)是由于例 3.3, 最后式(15.2.18)是由于条件概率的定义和概率值总是小于 1 的事实。

最后,注意到  $H$  预言机是均匀随机的,条件事件  $\text{AskG} | \overline{\text{AskH}}$ (即  $s^*$  没有被提问过的条件下  $r^*$  被提问这一事件)发生的概率只有  $2^{-k_0}$ 。我们还从式(15.2.9)知道  $\text{DBad}$  的概率也在  $2^{-k_0} + 2^{-k_1}$  这一数量级上。不等式(15.2.15) ~ (15.2.18)表明

$$\text{Prob}[\text{AskH}] \geq \text{Adv} - (2^{-k_0+1} + 2^{-k_1})$$

因此,如果  $\text{Adv}$  是关于  $k$  的不可忽略函数,那么  $\text{Prob}[\text{AskH}]$  也是。

现在,我们可以明显看出如果攻击者可以在 IND-CCA2 模式下攻破由随机预言机实现的 RSA-OAEP,那么它就可以对 RSA 函数部分求逆:以类似的优势找到  $s^*$ 。而对 RSA 函数的部分求逆确实能导致完全求逆。下面让我们看看这是为什么。

#### 15.2.4 对 RSA-OAEP 的补救工作

15.2.3.4 节中用到的数学知识确实对 RSA-OAEP 的挽救工作起了很重要的作用。可是,最初的挽救尝试并没利用到它。

##### 15.2.4.1 Shoup 的最初尝试

幸运的是,RSA 函数的可展性与基于 OAEP 变换的 Feistel 密码的可展性并不相像(回顾

图 15.2 以及我们对这两个结构之间代数性质不同的讨论)。具有讽刺意味的是,这两个结构之间代数性质(或可展性质)最大的不同使得 Shoup 可以证明 RSA-OAEP 是 IND-CCA2 安全的,只要 RSA 加密指数非常小:如果  $N$  是 RSA 模数,那么他的证明要求

$$k_0 \leq (\log_2 N)/e \quad (15.2.19)$$

让我们看一下为什么。

回顾我们的分析得出的结论:如果由询问密文  $c^*$  在式(15.2.5)中定义的  $s^*$  没有向预言机  $H$  提问,那么该归约在统计上是正确的。而该归约不正确的惟一情况是当  $s^*$  向  $H$  提问时。对于这种情况,我们还没有考虑 Simon 会如何反应。

Shoup 发现当  $s^*$  是 Simon 的  $H$  表中的一个  $(k - k_0)$  比特串时,Simon 可以对 RSA 问题解如下方程:

$$(X + 2^{k_0} I(s^*))^e \equiv c \pmod{N} \quad (15.2.20)$$

这里  $I(x)$  是串  $x$  的整数值。假设  $X < N^{1/e}$ , 用 Coppersmith 算法[83]我们可以在关于  $N$  的长度的多项式时间内解这个方程。因为  $X$  是大小为  $2^{k_0}$  量级的一个量,再由式(15.2.19)的限制,满足条件  $X < N^{1/e}$ 。

这样,当 Simon 被给定一条密文  $c$  要求解密服务时,一旦解密  $c$  失败他就使用 15.2.3.1 节中介绍的方法,使用在他的  $H$  表中的每个元素试图解式(15.2.20)求得  $X$ 。如果所有这些尝试都失败了,Simon 就拒绝  $c$ 。否则,解为  $X = I(t^*)$ ; 因为知道  $s^*$  和  $t^*$ , Simon 就可以像平常一样解密  $c$ 。

因此,在 RSA-OAEP 的这种情况下,即对于加密指数满足式(15.2.19)的情况,向  $H$  提问  $s^*$  早已帮助 Simon 对  $c^*$  求了逆。现在的问题是,  $e$  多大时才能满足式(15.2.19)? 对于 RSA-OAEP 的标准安全参数设置,我们有  $N > 2^{1024}$ ,  $k_0 = 160$  (为了使  $2^{-k_0}$  可忽略),从而  $e \leq \frac{1024}{160} = 6.4$ 。所以,在这个背景下,只有  $e = 3, e = 5$  是加密指数的可能取值( $e$  必须与偶数  $\phi(N)$  互素)。虽然使用这么小的指数我们就可以获得 RSA-OAEP 的可证明安全性,但是在 Coppersmith 的研究之后,人们已经广泛认识到对于 RSA 加密不能使用这么小的指数。

因为对  $k_0$  的标准安全参数设置已经接近下限,而且  $N$  的大小也不能大幅度增加,所以几乎不可能对任何更大的  $e$  使用这个归约。

#### 15.2.4.2 Fujisaki 等人的完全补救

又一次非常幸运,在 Shoup 的分析之后不久, Fujisaki 等人[116]做了更进一步的研究,并且发现了对一般的加密指数求逆 RSA 函数的方法。

对于 RSA-OAEP,  $s^*$  是  $c^*$  原像的很大一部分,向 Simon 泄漏很大部分的  $s^*$  确实泄漏得太多了。因为  $s^*$  有  $k - k_0$  比特,而且  $k > 2k_0$ , 超过  $c^*$  原像一半的比特(高位比特)被泄漏。给定原像的这么多比特, Fujisaki 等应用了著名的格技术,对任意大的  $e$  可以从下面的方程解出  $T = I(t^*)$ :

$$(2^{k_0} I(s^*) + T)^e \equiv c^* \pmod{N} \quad (15.2.21)$$

回忆给定一个 1 比特 RSA 预言机(“RSA 奇偶预言机”,回顾 9.2 节),我们研究过一个算法(算法 9.1),它应用一个 1 比特预言机  $\log_2 N$  次就可以对 RSA 函数求逆。完全相同的原理在这里也适用: $A$  实际上是一个“RSA 一半或更大分组预言机”,因为  $s^*$  拥有超过  $c^*$  原像一半的比

特。使用 Fujisaki 等人的算法, Simon 使用  $\mathcal{A}$  两次就可以得到原像部分信息的两个相关分组(一半或更大分组)。可以在式(15.2.21)中使用两个分组求出两个比  $\sqrt{N}$  小的未知整数。这两个整数之一就是  $T(t^*)$ , 因此 Simon 对 RSA 函数求了逆。

因为 Simon 必须使用  $\mathcal{A}$  两次, 所以他必须通过攻击游戏与  $\mathcal{A}$  进行两次归约: 一次给  $\mathcal{A}$  提供  $c^*$ , 另一次给  $\mathcal{A}$  提供  $\bar{c}^* = c^* \alpha^e \pmod{N}$ ,  $\alpha \in \mathbb{Z}_N^*$ , 是随机的。相应的  $s^*$  和  $\bar{s}^*$  分别在他的  $H$  表和  $\bar{H}$  表中。令  $q = \max(\#(H \text{ 表}), \#(\bar{H} \text{ 表}))$ 。由这两个表, Simon 只需要处理不超过  $q^2$  对  $(t, \bar{t})$ 。这些对中有一对可以使 Simon 得到式(15.2.21)的两个正确方程, 从而求得 RSA 函数的逆, 除非他选择了一个坏的  $\alpha$ , 但是该事件的概率非常小(当  $k \gg 2k_0$  时它发生的概率是可忽略的, 而对于 RSA-OAEP 恰有  $k \gg 2k_0$ )。因为解式(15.2.21)的两种情况可以在  $O_B((\log_2 N)^3)$  时间内完成, 所以 Simon 可以在

$$2\tau + q^2 \times O_B((\log_2 N)^3) \quad (15.2.22)$$

时间内求得 RSA 函数的逆,  $\tau$  是  $\mathcal{A}$  对 RSA-OAEP 执行 IND-CCA2 攻击的时间界。

对于填充参数的设置, RSA-OAEP 还有两个变形。它们是 PKCS #1 的第 2 版及更高版本 [232] 和 SET [261]。在这两个变形中, 已知的数据块  $s^*$  被放在明文块中的不同位置。因为  $s^*$  足够大(比一半大得多), 所以最多运行  $\mathcal{A}$  两次, 就可以很容易地完成根的求解。因此, 对 Fujisaki 等技术的一个变形仍然可以应用到这两个变形中。

在这种方式下, RSA-OAEP 及其变形仍然具有 IND-CCA2 模式下的可证明安全性。

最后, 我们指出同样的结果也适用于 Rabin-OAEP。因为对 Rabin 函数的求逆, 即在任意点求模  $N$  的平方根, 意味着分解  $N$ , 该证明可以通过 Simon 应用定理 6.17(iii) (见 6.6.2 节) 完成: 选择一个随机值  $x$ , 令  $c^*$  为  $x$  的未填充 Rabin 加密。因为分解假设比 RSA 假设弱, 所以 Rabin-OAEP 的安全性结果比 RSA-OAEP 的好。

### 15.2.5 RSA-OAEP“归约为矛盾”的严谨性

RSA-OAEP 方案是非常有效的, 但是我们不能认为它的“归约为矛盾”也是如此。现在我们讨论这个问题。

表达式(15.2.22)给出了 Simon 仿真器应用  $\mathcal{A}$  两次在任意点处对 RSA 函数求逆所需的时间。该表达式有一个二次项  $q^2$ ,  $q$  是在 Simon 使用的每次实例中,  $\mathcal{A}$  可以对  $H$  进行的随机预言提问的次数。

注意, 随机预言机理想化了可以有效计算的杂凑函数。对于献身的攻击者来说, 我们必须合理地允许它进行譬如  $2^{50}$  次杂凑函数计算。所以, 考虑  $q \approx 2^{50}$  是合理的。这样, 式(15.2.22)中的二次项  $q^2$  就意味着 Simon 求逆 RSA 函数的时间是

$$2^{100} \cdot O_B((\log_2 N)^3)$$

现在回顾 4.6 节因子分解技巧的状况, 式(4.6.1)用数域筛法(NFS)分解  $N$  的时间复杂度表达式。对于通常的大小  $|N| = 1024$ , 式(4.6.1)提供了一个  $2^86$  数量级的值。因此, 由  $2^{100} \cdot O_B((\log_2 N)^3)$  给出的矛盾根本没有意义, 因为使用 NFS 方法, Simon 就可以不使用  $\mathcal{A}$  在少得多的时间内对基于 1024 比特模数的 RSA 函数求逆。所以这个“归约为矛盾”的证明对于 1024 比特的 RSA 模数来说不是有效的。



因为目前认为 1024 比特的 RSA 模数对很多安全应用都是安全的,所以 RSA-OAEP 安全性证明的无效性表明该归约作为一个二次多项式并不令人满意。

“归约为矛盾”的证明对更大的 RSA 模数是有效的,譬如,它对 2048 比特的模数就刚好有效,因为此时式(4.6.1)会产生  $2^{116}$  数量级的一个值。

### 15.2.6 对随机预言机模型的批评

Canetti、Goldreich 和 Halevi 对基于 ROM 的安全性证明持相当否定的态度[65,66]。他们证明存在签名和加密方案,在 ROM 下是可证明安全的,但在实际中不会有任何安全的实现。他们的基本思想是设计一些危险的方案,这样的方案通常和一个签名或加密方案一样工作正常。但是,一旦满足某个条件(最基本的,当发现不随机时),这些方案就会变得很危险,如果是签名方案,它们会输出保密的签名密钥;如果是加密方案,它们会输出明文消息。

显然,当我们证明这样的危险方案在 ROM 下的安全性时,由于假设签名或加密的明文是均匀随机的(当然仅仅是为了使用 ROM),所以可以证明。但是,在实际应用的真实世界中,不存在均匀随机的明文,所以任何实际实现都显然不是安全的。

他们构造这些危险方案的步骤非常复杂。对此比较感兴趣的读者可以参阅[66]。

但是,非常有趣的是,在优美而且令人信服的科学论证之后,我们发现这三个作者关于随机预言机方法的作用得到了大不相同的结论。他们决定用三个分离的、也就是每人一个的结论,以最矛盾的方式给出这些结论的不一致性。

Canetti 的结论([66]的 6.1 节)揭示了这三个结论中最具批评意味的观点。他认为随机预言机模型是一个糟糕的抽象概念,导致了归约到困难问题的丧失(即它失去了“归约为矛盾”的优美思想)。他还认为识别随机预言机的任何有用的、具有特殊目的的性质可以是另一个可选的研究方向。

Goldreich 的结论([66]的 6.2 节)是三个批评结论中最轻的。他把与 ROM 有关的问题看做是不完全性:在随机预言机实例中可能无法排除因为某些缺陷造成的不安全性。因此,他建议在现在的公开工作中,不要使用在 ROM 下的安全性证明(我们把这句话解释为:这些证明不应该看做是真正的证明)。但是,他还有一个很乐观的底线:该模型有它的价值,在密码学方案的健全性测试中起到了试验台的作用。他还希望将来这个模型可以体现更多值得借鉴的价值。

Halevi 的结论([66]的 6.3 节)涉及到一个似乎不是不可忽略概率的事件。他认为这个方法的暂时成功完全是因为运气:“目前所有在随机预言机模型下被证明安全的方案碰巧也在真实世界中安全,这一点说不出原因。”他的底线是:今天的标准应该在具有 ROM 证明的方案之中,而不是在那些不具有这些证明的方案之中。这毕竟是一个相当乐观的底线。

### 15.2.7 作者对随机预言机模型价值的观点

本书的作者对于基于随机预言机模型的安全性证明的价值有自己的观点。为了保持目前我们已学习过的本章内容的客观性,我们只限于研究 RSA-OAEP 加密算法的情况。

对 RSA-OAEP 基于 ROM 的安全性证明本质上揭示了以下事实:

如果填充方案确实是一个随机函数,那么该填充导致 OAEP 输出的结果是理想世界中的一条“明文”:它在 RSA 函数的明文空间内具有均匀随机分布。所以,在 9.2 节

中,对于 RSA 函数在理想世界中的强度,我们的研究表明攻破 IND-CCA2 安全性的最简单方法是首先解决 RSA 问题然后解密。

所以,基于 ROM 的证明暗示着:对于使用真实世界中的杂凑函数(而不是随机预言机)的基于填充的加密方案,它们对攻击最脆弱的一个环节是在方案中使用的杂凑函数。为了获得对基于填充的加密方案的高度信任,我们应该注意杂凑函数的设计及其输入的随机性。

从这个观点出发,我们认为对于基于 ROM 的安全性证明技术,它的重要性在于它指出了细心设计时应当特别注意的地方。

### 15.3 Cramer-Shoup 公钥密码体制

另一个可证明 IND-CCA2 安全的、在实际中有效的著名公钥密码体制是 Cramer-Shoup 公钥密码体制[84],由其发明者 Cramer 和 Shoup 的名字命名。

#### 15.3.1 在标准困难性假设下的可证明安全性

我们看到形式化可证明安全性的一般方法是:将对密码方案所谓的攻击“归约”到对一个著名困难问题的解(即利用所谓的成功攻击者,把它当做一个黑盒,解决著名的困难问题)。我们希望这种“归约为矛盾”的证明具有下列两个重要特性。

**性质 15.1** 对“归约为矛盾”所希望有的性质

- i) 归约应该是有效的;理想情况下,对密码方案的一个所谓成功攻击者应该可以在类似于进行该攻击的努力下,解决该方案所基于的困难问题。
- ii) 方案安全所需要的困难性假设应该越弱越好;理想地,对于一个基于陷门单向函数(OWTF,注意 OWTP 是 OWTF 的一种特殊情况)的公钥加密方案,为了使该方案可证明安全,惟一的假设应该是该陷门单向函数的难解性。

性质 15.1(i)在实际中很重要:一个有效的归约,即使是多项式时间的,也可能根本不能给出攻击和困难问题之间的实际关系。例如,如果一个归约关系是次数为 8 的多项式,此时的安全参数和通常一样,是 1024,那么该归约的时间复杂度是  $1024^8 = 2^{80}$  的数量级。在这种归约下,当攻击者有一个非常有效的攻击算法在  $10^{-6}$  秒攻破方案时,使用该攻击者的归约却只能在 380 亿年内解决一个难问题!这样的可证明安全性不仅肯定没有用,而且不能构成量化一个数学证明的任何矛盾:已知求解该著名困难问题的方法所需要的时间可能也远远小于由该归约给出的数字!事实上,正如我们在 15.2.5 节中看到的,对于某些使用大小相当标准的安全参数的应用,即使是由 2 次多项式度量的归约,也已经可以看做是无效的了。

有人可能会认为希望具有的性质 15.1(ii)在实际中不太重要,因为它似乎只是与数学证明中的一般准则一致:如果假设的减弱并不限制证明的推导,那么该证明应该只是基于这个减弱的假设。得到一个漂亮的证明当然是我们动机的重要部分,所以从实践来看,性质 15.1(ii)就更重要了。在设计密码学系统时,性质 15.1(ii)尤其重要;当我们使用比较实际而且容易得到的密码学构造时,越弱的假设就越容易满足,因此,使用较弱假设的密码学系统就比使用较强假设的密码学系统提供了更高的安全可信度。

我们知道基于 ROM 对 RSA-OAEP 的证明不满足性质 15.1(i) 的理想情况, 因为对于标准的 RSA 模数, 该归约是不够严谨的。但它却不<sup>①</sup>能很好地满足性质 15.1(ii), 这是因为该证明不仅需要 RSA 函数的困难性(RSA 假设, 假设 8.3), 而且需要一个强得多的假设: OAEP 构造中使用的杂凑函数必须具有随机预言机的性质。这个假设非常强, 事实上, 它强得很不合理: 正如我们在 10.3.1.2 节中讨论过的, 在真实世界中不存在随机预言机; 因此, 这个假设在数学上是不能满足的。从实际讲, 我们从 RSA-OAEP 的证明所获得的, 事实上是在 RSA-OAEP 方案的构造中必须使用高质量的杂凑函数。遗憾的是, 这不是绝对可信的, 也不是数学证明应该提供的。

对于公钥密码体制, 如果安全性的形式化证明只依赖于体制所基于的 OWTP 的困难性, 那么该证明就称为是在(一些)标准困难性假设下的证明。这种证明建立了真实世界中的安全性: 它证明一个密码体制是不可攻破的, 只要它所基于的(这些)困难性假设不能攻破。

有很多基于标准困难性假设可证明安全的(在 IND-CCA2 模式下)密码体制, 例如 Dolev 等人的 NM-CCA2(等价于 IND-CCA2)安全的方案[101]。但是, 正如我们在 14.5.3 节中讨论过的, 方案中需要使用的 NIZK 证明使其在实际应用没有什么吸引力。

Cramer-Shoup 体制[85]是第一个实际有效而且在标准困难性假设下可证明 IND-CCA2 安全的公钥密码体制。我们还会看到该方案有一个严谨的“归约为矛盾”的安全性证明: 线性归约。所以 Cramer-Shoup 体制满足希望具有的性质 15.1 中的理想性质。

现在我们介绍 Cramer-Shoup 体制。

### 15.3.2 Cramer-Shoup 体制

Cramer-Shoup 公钥密码体制是对语义安全 ElGamal 加密方案(见 14.3.5 节)的一个 CCA2 改进。与语义安全的 ElGamal 加密方案情况相同, Cramer-Shoup 体制安全性所基于的标准困难性假设是判定 Diffie-Hellman(DDH)假设。读者可以回顾 13.3.4.3 节定义 13.1 中的 DDH 问题, 以及 14.3.5 节假设 14.2 中的 DDH 假设。

算法 15.1 具体介绍了 Cramer-Shoup 体制。

#### 算法 15.1 Cramer-Shoup 公钥密码体制

##### 密钥参数

假设  $G$  是一个具有大素数阶  $q$  的阿贝尔群。明文空间是  $G$ ;

(\* 我们假设存在编码方案将任意的明文编码为  $G$  中的一个比特串, 而且可以解码; 给定  $\text{desc}(G)$ , 这样的编码方案很容易实现, 例如见 14.3.5 节 \*)

为了建立用户的密钥材料, 用户 Alice 执行下列步骤:

1. 选择两个随机元  $g_1, g_2 \in {}_U G$ ;
2. 选择五个随机整数  $x_1, x_2, y_1, y_2, z \in {}_U [0, q)$ ;
3. 计算  $c \leftarrow g_1^{x_1} g_2^{x_2}$ ,  $d \leftarrow g_1^{y_1} g_2^{y_2}$ ,  $h \leftarrow g_1^z$ ;
4. 选择一个密码学杂凑函数  $H: G^3 \mapsto [0, q)$ ;

<sup>①</sup> 此处原文少了“not”。

5. 公开 $(g_1, g_2, c, d, h, H)$ 作为公钥,保留 $(x_1, x_2, y_1, y_2, z)$ 为私钥。

#### 加密

为了发送一条秘密消息  $m \in G$  给 Alice, 发送者 Bob 选择一个随机整数  $r \in_U [0, q)$ , 并计算

$$u_1 \leftarrow g_1^r, u_2 \leftarrow g_2^r, e \leftarrow h^r m, \alpha \leftarrow H(u_1, u_2, e), v \leftarrow e^r d^m$$

密文是 $(u_1, u_2, e, v)$ 。

#### 解密

为解密密文 $(u_1, u_2, e, v)$ , Alice 执行下列步骤:

1.  $\alpha \leftarrow H(u_1, u_2, e)$ ;
2. 输出  $\begin{cases} m \leftarrow e/u_1^\alpha & \text{若 } u_1^{x_1+\gamma_1\alpha} u_2^{x_2+\gamma_2\alpha} = v \\ \text{REJECT} & \text{其他情况} \end{cases}$

不难看出密文 $(u_1, e)$ 的这一部分正好是语义安全 ElGamal 体制的密文对。由定理 14.2 (见 14.3.5 节), 我们已经知道 Cramer-Shoup 体制在 DDH 假设下是 IND-CPA 安全的。

与其他 CCA2 安全的加密方案一样, 解密过程有一个数据完整性验证步骤。假设密文在发送给 Alice 的途中没有被修改, 我们有

$$u_1^{x_1+\gamma_1\alpha} u_2^{x_2+\gamma_2\alpha} = (u_1^{x_1} u_2^{x_2}) (u_1^{\gamma_1\alpha} u_2^{\gamma_2\alpha}) = (g_1^{x_1} g_2^{x_2}) (g_1^{\gamma_1\alpha} g_2^{\gamma_2\alpha}) = c^r d^m = v$$

一旦通过这个消息完整性验证, 剩下的解密过程与语义安全的 ElGamal 体制相同。后面我们将看到这个消息完整性验证步骤非常有效: 它确实能够阻止不使用该加密过程就构造合法密文的可能性。

读者可能会有下述疑问:

“为什么该方案的安全性只依赖于 DDH 假设? 既然消息完整性验证使用一个杂凑函数, 那么为什么该方案的安全性和杂凑函数的某个性质如随机预言机性质没有关系呢?”

该方案中使用的杂凑函数当然不能是弱杂凑函数。但是, 我们应该注意到在数据完整性验证中使用的安全性服务只是杂凑函数的单向性, 而没有必要使用随机预言机性质。例如, 杂凑函数  $H(x)$  可以用同一个群  $G$  中的  $g^x$  实现, 这样, 我们只要用离散对数(DL)问题(见 8.4 节中的定义 8.2)的单向性。相关的困难性假设是离散对数(DL)假设(见 8.4 节中的假设 8.2), 这个假设不仅是标准的, 而且在同一个群中比 DDH 假设还弱: 即如果我们使用  $G$  中的 DDH 假设, 那么 DL 假设在  $G$  中必然也成立。正是从这一点出发, 我们说该方案的安全性只依赖于 DDH 假设。相反, 在 Shoup 对  $f$ -OAEP 的攻击(见 15.2.3.3 节)中, 我们看到  $f$ -OAEP 的安全性证明不能只依赖于单向性, 这包括 OAEP 构造中使用的杂凑函数, 以及所基于的困难问题。

#### 15.3.2.1 性能

表面看来, Cramer-Shoup 体制与 ElGamal 体制相比, 似乎使用了非常大的密钥以及更多的指数, 但是, 进一步研究就会发现这个差别并不大。

该体制的公钥由  $G$  中的五个元素组成, 比 ElGamal 的两个元素有所增加。密文的规模是  $G$  中的四元组, 是 ElGamal 的两倍。加密需要“五个”指数运算(但是其实是四个, 我们很快会看到),

也比 ElGamal 的两个有所增加。解密需要“三个”(实际上是两个)指数运算,比 ElGamal 的两个有所增加。

现在我们解释为什么加密(解密)只需要其中的四个(两个)指数运算,而不是方案中说明的五个(三个)。这是因为公式  $g^x h^y$  中两个幂的乘积可以用一个指数运算的代价计算。算法 15.2 具体说明了这种算法。不难看出该算法在  $\max(|x|, |y|)$  步熟知的“平方与乘积”运算内会结束,并输出正确的结果。注意在该算法中,事实上是在对 Cramer-Shoup 体制的整个介绍过程中,我们省略了对群运算的陈述。实际上,  $G$  可以是任何 DDH 假设成立的阿贝尔群。

### 算法 15.2 幂的乘积

输入  $g, h \in A$ ,  $A$  是一个代数结构;  $x, y$ : 在区间  $(0, \#A)$  内的整数;  $\text{Exp}(u, z)$ : 单个指数运算, 返回  $u^z$ ; (\* 例如, 使用算法 4.3 的  $\text{Exp}$  \*)

输出  $g^x h^y$ 。

1. if( $|x| > |y|$ )

{

$u \leftarrow \text{Exp}(g, x \bmod 2^{|x|-|y|})$ ;

(\* 指数运算使用  $x$  的  $|x| - |y|$  个低位比特 \*)

$x \leftarrow x \div 2^{|x|-|y|}$  (\* “ $\div$ ”: 整数除法; 该操作去掉  $x$  的  $|x| - |y|$  个低位比特, 使得  $|x| = |y|$  \*)

}

2. if( $|y| > |x|$ )

{

$u \leftarrow \text{Exp}(h, y \bmod 2^{|y|-|x|})$ ;

$y \leftarrow y \div 2^{|y|-|x|}$

}

3.  $v \leftarrow gh$ ; (\* 这以下  $|x| = |y|$  \*)

4. while( $x \neq 0$ ) do

{

(a)  $u \leftarrow u^2$ ;

(b) if( $x \bmod 2 = 1 \wedge y \bmod 2 = 1$ )  $u \leftarrow uv$ ;

(c) if( $x \bmod 2 = 1 \wedge y \bmod 2 = 0$ )  $u \leftarrow ug$ ;

(d) if( $x \bmod 2 = 0 \wedge y \bmod 2 = 1$ )  $u \leftarrow uh$ ;

(e)  $x \leftarrow x \div 2$ ;  $y \leftarrow y \div 2$  (\* 丢弃最低位比特 \*)

}

5. return( $u$ ).

(\* 全部“平方与乘积”运算的次数:  $\max(|x|, |y|)$  \*)

对 Cramer-Shoup 密码体制的性能研究之后, 我们可以得出如下结论: 从通信带宽和计算两方面看, Cramer-Shoup 体制的代价均大约是 ElGamal 体制的两倍。

### 15.3.3 安全性证明

对于具有适于应用安全性的 Cramer-Shoup 体制,如果读者只想知道怎样使用它来加密,那么算法 15.1 提供了足够的关于“知其然”信息,所以可以直接跳至 15.4 节。从这里一直到 15.4 节之前的内容是关于“知其所以然”的材料:回答为什么 Cramer-Shoup 体制具有适于应用的安全性。我们力图以直观的方式给出答案。

决定跟着我们“知其所以然”路线的读者不需要任何高深的数学知识就可以理解对 Cramer-Shoup 体制安全性的证明方法。基本了解我们在 15.2 节中介绍的群论再加上线性代数的基本知识(我们会陈述用到的事实)就已经足够了。

对 Cramer-Shoup 体制的安全性证明使用“归约为矛盾”的方法以达到形式化可证明安全性:将一个由基本困难性假设所支持的困难问题“归约”为一个所谓的 IND-CCA2 攻击。在 Cramer-Shoup 体制中,困难问题如下:

假设  $G$  是一个具有大素数阶  $q$  的群,  $(g_1, g_2, u_1, u_2) \in G^4$  是任意的一个四元组,  $g_1 \neq 1, g_2 \neq 1$ 。回答问题:  $(g_1, g_2, u_1, u_2)$  是 Diffie-Hellman 四元组吗? 即是否存在整数  $a, b \in [0, q)$ , 满足

$$g_2 = g_1^a, u_1 = g_1^b, u_2 = g_1^{ab} \quad (15.3.1)$$

因为  $G$  是素阶群,  $g_1 \neq 1$  是  $G$  的一个生成元(推论 5.3), 因此总是存在整数  $a, b \in [0, q)$  满足式(15.3.1)中的前两个方程。这就是我们为什么只对第三个方程加问号的原因。无须验证, 式(15.3.1)中三个方程成立等价于

$$\log_{g_1} u_1 = \log_{g_2} u_2 \pmod{q}$$

由判定 Diffie-Hellman 假设(假设 14.2), 该问题对一般阿贝尔群都是困难问题。

#### 15.3.3.1 证明安全性技术的顶层描述

假设存在一个攻击者  $A$  可以以一个不可忽略的概率在 IND-CCA2 模式下攻破 Cramer-Shoup 体制。我们要构造一个有效的归约算法, 使我们的特殊代理 Simon 仿真器回答 DDH 问题。

Simon 的输入是任意的一个四元组  $(g_1, g_2, u_1, u_2) \in G^4$ , 这里  $g_1 \neq 1, g_2 \neq 1$ 。这个四元组可能是 Diffie-Hellman 四元组, 如果确实如此, 我们记为  $(g_1, g_2, u_1, u_2) \in \mathbf{D}$ ; 否则我们记为  $(g_1, g_2, u_1, u_2) \notin \mathbf{D}$ 。

使用输入值, Simon 可以构造公钥  $PK = (g_1, g_2, c, d, h, H)$  让  $A$  使用, 并且在他与  $A$  进行 IND-CCA2 攻击游戏时, 他也可以在收到  $A$  的要求时构造一条询问密文  $C^* = (u_1, u_2, e, v)$ , 它是选择明文  $m_b \in_U \{m_0, m_1\}$  的加密( $m_0, m_1$  由  $A$  选择, 但比特  $b$  对  $A$  保密)。

询问密文  $C^*$  有下面两个性质:

- i) 如果  $(g_1, g_2, u_1, u_2) \in \mathbf{D}$ , 那么  $C^*$  是一条有效的 Cramer-Shoup 密文, 它是在公钥  $PK$  下对  $m_b$  的加密。我们将在 15.3.3.3 节和 15.3.3.4 节中看到  $C^*$  的有效性。而且, 不管是否使用给定的公钥,  $A$  都可以获得密码分析训练课程, 这将由 Simon 精确地仿真。在 15.3.3.5 节中我们会看到仿真分析训练课程的精确性。所以, 在这种情况下, Simon 要求  $A$  完全发挥它的攻击优势。



- ii) 如果  $(g_1, g_2, u_1, u_2) \notin \mathbf{D}$ , 那么询问密文  $C^*$  是  $m_b$  在香农信息论安全性意义下的加密 (即完善加密, 见 7.5 节), 即密文将在整个密文空间中均匀分布。我们会在 15.3.3.4 节中看到香农的完善加密。而且, 我们在 15.3.3.5 节中还会看到香农完善加密的性质不会因为给  $\mathcal{A}$  提供密码分析训练课程而泄密。所以, 在这种情况下,  $\mathcal{A}$  无论如何也不会有任何优势!

正是这两种情况下各自优势的差别使得  $\mathcal{A}$  很好地教会了 Simon 回答 DDH 问题。现在我们构造“归约为矛盾”。

### 15.3.3.2 归约

该归约包括以下步骤:

1. 输入  $(g_1, g_2, u_1, u_2) \in G^4$ , Simon 构造 Cramer-Shoup 体制的一个公钥, 将其发送给  $\mathcal{A}$ ; 构造公钥的方法在 15.3.3.3 节中描述。
2. Simon 为  $\mathcal{A}$  提供在询问之前需要的密码分析训练课程; Simon 仿真  $\mathcal{O}$  解密的方法在 15.3.3.5 节中描述。
3. Simon 从  $\mathcal{A}$  收到一对选择明文  $m_0, m_1$ , 投掷一个公平的硬币  $b \in_U \{0, 1\}$ , 加密  $m_b$  构造询问密文  $C^*$ , 将  $C^*$  发送给  $\mathcal{A}$ ; Simon 仿真  $\mathcal{O}$  加密的方法在 15.3.3.5 节中描述。
4. Simon 通过仿真  $\mathcal{O}$  的解密过程继续为  $\mathcal{A}$  提供询问之后它需要的训练课程。
5. Simon 最后收到  $\mathcal{A}$  对比特  $b$  的有根据猜测; 这时, Simon 就可以回答问题:  $(g_1, g_2, u_1, u_2) \in \mathbf{D}$  还是  $(g_1, g_2, u_1, u_2) \notin \mathbf{D}$ 。

图 15.4 给出了该归约的图示说明。它是在 IND-CCA2 攻击者  $\mathcal{A}$  和 Simon 仿真器之间进行的一个攻击游戏。Simon 控制了  $\mathcal{A}$  的所有通信途径, 使得  $\mathcal{A}$  只能与 Simon 交互。对于 Simon, 这个攻击游戏是一个仿真的游戏。但是, 正如我们即将看到的, 因为仿真的质量非常完美, 所以  $\mathcal{A}$  不能辨别真正的攻击游戏和这个仿真的攻击游戏。

### 15.3.3.3 公钥构造

使用输入的四元组  $(g_1, g_2, u_1, u_2) \in G^4$ , Simon 如下构造公钥: 他选择

$$x_1, x_2, y_1, y_2, z_1, z_2 \in_U [0, q)$$

并计算

$$c \leftarrow g_1^{x_1} g_2^{x_2}, d \leftarrow g_1^{y_1} g_2^{y_2}, h \leftarrow g_1^{z_1} g_2^{z_2} \quad (15.3.2)$$

Simon 再选择一个密码学杂凑函数  $H$ 。  $\mathcal{A}$  要使用的公钥是  $(g_1, g_2, c, d, h, H)$ 。

Simon 使用的私钥是  $(x_1, x_2, y_1, y_2, z_1, z_2)$ 。

读者可能已经注意到该公钥的一部分, 即  $h$  与算法 15.1 中描述的不同。我们现在解释这里没有问题。首先, 我们证明对于 Simon 构造的公钥,  $h$  这一分量完全是有效的。

对  $h = g_1^{z_1} g_2^{z_2}$ , 其中  $g_1 \neq 1$ , 注意  $g_1$  是  $G$  的一个生成元 (见推论 5.3), 因此对某个  $w \in [0, q)$  有  $g_2 = g_1^w$ ; 这样对  $z \equiv z_1 + wz_2 \pmod{q}$  就有

$$h = g_1^{z_1 + wz_2} = g_1^z \quad (15.3.3)$$

所以,  $h$  确实服从算法 15.1 的密钥建立过程。

读者可能还会有下面的疑问：

“既然 Simon 不知道  $w = \log_{g_1} g_2 \pmod{q}$ ，那么我们在后面的解密过程中怎么使用

$$z \equiv z_1 + wz_2 \pmod{q}?$$

在 15.3.3.5 节中，我们将看到对于公钥的任意有效密文，Simon 确实都可以正确地使用  $z \equiv z_1 + wz_2 \pmod{q}$  作为“正常”的解密指数，即使他不知道  $z$ 。

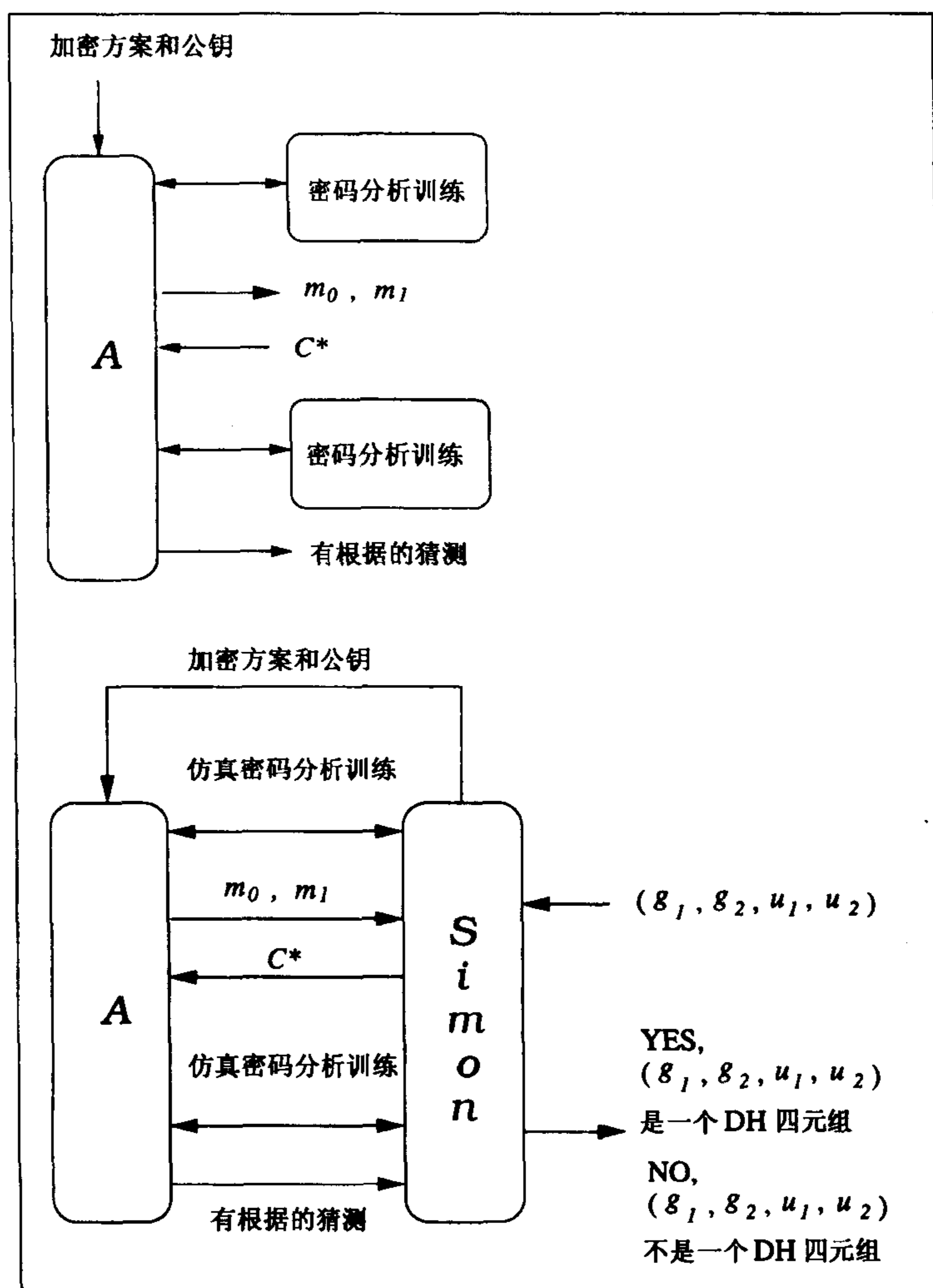


图 15.4 从 DDH 问题到对 Cramer-Shoup 体制的攻击的归约

#### 15.3.3.4 加密过程的仿真

一旦从  $A$  收到两个选择明文  $m_0, m_1$ ，Simon 就投掷一个公平硬币  $b \in_U \{0, 1\}$ ，并如下加密  $m_b$ ：

$$e = u_1^{z_1} u_2^{z_2} m_b, \alpha = H(u_1, u_2, e), v = u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha}$$

询问密文  $C^*$  是  $(u_1, u_2, e, v)$ 。

读者可能会再次注意到这个加密过程也和正常的加密过程不同,正常的加密中  $e$  是如下计算的,对某个  $r \in [0, q)$ ,  $e = h^r m_b$ 。

但是,这根本不会产生任何问题。相反,这个不同对安全性证明却至关重要。我们通过下面两种情况解释这个关键性:

- i)  $(g_1, g_2, u_1, u_2) \in \mathbf{D}$ 。在这种情况下,因为存在  $r \in [0, q)$ ,使得  $u_1 = g_1^r, u_2 = g_2^r$ ,所以有  $u_1^{z_1} u_2^{z_2} = (g_1^r)^{z_1} (g_2^r)^{z_2} = (g_1^{z_1} g_2^{z_2})^r = h^r$ 。因此仿真的加密确实是一个给定公钥下的有效 Cramer-Shoup 加密。这也正是我们所期望的,因为我们希望在这种情况下  $\mathcal{A}$  能发挥它全部的攻击优势。
- ii)  $(g_1, g_2, u_1, u_2) \notin \mathbf{D}$ 。在这种情况下,存在整数  $r_1, r_2 \in [0, q), r_1 \not\equiv r_2 \pmod{q}$ ,使得  $u_1 = g_1^{r_1}, u_2 = g_2^{r_2}$ 。因为  $g_1 \neq 1$  是  $G$  的一个生成元,所以存在  $\log_{g_1} g_2, \log_{g_1} h, \log_{g_1}(e/m_0)$  和  $\log_{g_1}(e/m_1)$ 。

为了更清楚地解释,我们可以考虑  $\mathcal{A}$  现在(即只在这种情况下)是计算上无界的。给定询问密文  $C^*$  中的  $e$ ,具有无限计算能力的  $\mathcal{A}$  可以看到下列关于两个未知整数  $(z_1, z_2)$  的两个线性方程组:

$$\begin{pmatrix} 1 & \log_{g_1} g_2 \\ r_1 & r_2 \log_{g_1} g_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \log_{g_1} h \\ \log_{g_1}(e/m_0) \end{pmatrix} \pmod{q} \quad (15.3.4)$$

$$\begin{pmatrix} 1 & \log_{g_1} g_2 \\ r_1 & r_2 \log_{g_1} g_2 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \log_{g_1} h \\ \log_{g_1}(e/m_1) \end{pmatrix} \pmod{q} \quad (15.3.5)$$

由  $(g_1, g_2, u_1, u_2) \notin \mathbf{D}$ , 我们有  $r_1 \not\equiv r_2 \pmod{q}$ ; 同时注意到  $\log_{g_1} g_2 \not\equiv 0 \pmod{q}$  (因为  $g_2 \neq 1$ )。所以左边的矩阵是满秩的,即秩为 2,因此这两个方程组对于  $(z_1, z_2)$  都有惟一的整数解。 $\mathcal{A}$  无法验证这两种情况哪一个是正确的,这样,即使  $\mathcal{A}$  是计算无界的,它也完全不知道密文  $C^*$  到底是  $m_0$  还是  $m_1$  的加密。所以说,在这种情况下,  $C^*$  是  $m_b$  在香农的信息论安全性意义下的加密,因而  $\mathcal{A}$  无论如何也不会有优势!

我们必须指出,到目前为止,对情况(i)确切的 Cramer-Shoup 加密,或者对情况(ii)的香农的信息论安全加密,都还只是在 CPA 模式下成立。也就是说,如果  $\mathcal{A}$  是被动的,那么仿真加密的质量在两种情况下都成立。但是我们的攻击者并不是这么弱! 记住,  $\mathcal{A}$  甚至在收到询问密文之后还可以得到密码分析训练课程。譬如在情况(ii)中如果  $\mathcal{A}$  可以得到除了式(15.3.4)和式(15.3.5)以外的第三个线性方程组,那么作为 CCA2 训练课程的结果,我们可能就不能再说这个加密是香农信息论安全的了。

我们在下一节将看到,在 IND-CCA2 攻击者享有的整个密码分析训练课程中,如何保持这两种情况下仿真加密的质量。

### 15.3.3.5 解密过程的仿真

收到  $\mathcal{A}$  的密文  $C = (U_1, U_2, E, V)$  后, Simon 首先进行算法 15.1 中的数据完整性验证过程。如果测试结果为 YES,那么就认为该密文是有效的。然后 Simon 计算

$$m = E/(U_1^{z_1} U_2^{z_2}) \quad (15.3.6)$$

将  $m$  返回给  $A$  作为解密结果。如果该测试结果为 NO, 那么就认为该密文是无效的, Simon 将返回 REJECT 作为解密结果。

不久我们会陈述并证明定理 15.1, 该定理表明有效密文  $C = (U_1, U_2, E, V)$  意味着  $(g_1, g_2, U_1, U_2) \in \mathbf{D}$  的概率是  $1 - \frac{1}{q}$ 。所以存在  $R \in [0, q)$ , 使得  $U_1 = g_1^R, U_2 = g_2^R$ , 从而有

$$U_1^{z_1} U_2^{z_2} = (g_1^R)^{z_1} (g_2^R)^{z_2} = (g_1^{z_1} g_2^{z_2})^R = h^R. \quad (15.3.7)$$

这样我们就可以看出 Simon 在式(15.3.6)中执行的仿真解密是正确的, 除了一个可忽略的概率  $\frac{1}{q}$ 。这澄清了我们在 15.3.3.3 节结尾处留下的疑问: Simon 在没有“正常”秘密指数  $z \equiv z_1 + z_2 \log_{g_1} g_2 \pmod{q}$  的情况下怎样正确解密。

Simon 能够对有效的密文正确解密, 于是 Simon 可以为  $A$  提供密码分析训练课程, 这是  $A$  作为一个 IND-CCA2 攻击者应该得到的。

现在我们证明该密码分析训练课程不会危及到询问密文对  $m_b$  的完美隐藏, 这在 15.3.3.4 节中已经给出了。

对于  $A$  提交的任意有效密文, 返回的解密结果只能使  $A$  确信式(15.3.7)中由  $(U_1, U_2, h^R)$  定义整数对  $(z_1, z_2)$  的方式与用式(15.3.2)中第三个等式由公钥组成部分  $(g_1, g_2, h)$  定义  $(z_1, z_2)$  的方式是相同的。所以, 除了在公钥中已经给出的信息,  $A$  得不到任何关于  $(z_1, z_2)$  的信息。因此, 如果  $A$  提交有效的密文, 密码分析训练课程对它就是没有用的。

为了不浪费猜测前的密码分析训练机会,  $A$  必须提交满足  $(g_1, g_2, U_1, U_2) \in \mathbf{D}$  的密文  $C = (U_1, U_2, E, V)$ 。如果这样的密文通过了 Simon 的验证过程, 那么一个数字的解密结果会返回给  $A$ , 而且这个解密结果可能和询问密文有某种只有  $A$  知道的关系。因为我们假设  $A$  是非常聪明的, 所以在  $(g_1, g_2, U_1, U_2) \notin \mathbf{D}$  的情况下, 我们总是不能确认返回的解密结果与询问密文有何种关系。如果我们确信  $A$  有一个隐藏的关系, 那么就不能再像 15.3.3.4 节中在 CPA 模式下建立的结论那样, 声称对于情况(i)  $m_b$  的加密确实是在 Cramer-Shoup 方案下的加密, 或者对于情况(ii)  $m_b$  的加密是信息论安全的。

幸运的是, 如果  $A$  以某种方式得到了一条使得  $(g_1, g_2, U_1, U_2) \notin \mathbf{D}$  的密文  $(U_1, U_2, E, V)$ , 那么它立刻会得到 REJECT 的答案。这是由于定理 15.1, 对此我们马上就要陈述并证明。我们将看到, 拒绝的概率至少是  $1 - \frac{1}{q}$ 。注意在剩下的  $\frac{1}{q}$  概率中,  $A$  没有必要很麻烦地提交密文然后从 Simon 那里得到任何线索;  $A$  总是可以自己以  $\frac{1}{q}$  的概率正确猜测  $G$  中的任何东西, 因为  $G$  的大小只有  $q$ 。

这样,  $A$  就不能通过提交它希望不会被拒绝的坏密文来耍“小聪明”了。

构造一条坏密文并能逃避被拒绝的概率可以如下建立。

**定理 15.1** 假设  $(g_1, g_2, c, d, h, H)$  是素数  $q$  阶群  $G$  中 Cramer-Shoup 加密方案的一个公钥, 其

中  $g_1 \neq 1, g_2 \neq 1$ 。如果  $(g_1, g_2, U_1, U_2) \notin \mathbf{D}$ , 那么不管使用什么算法, 解决下述问题的成功概率不超过  $\frac{1}{q}$ :

输入: 公钥  $(g_1, g_2, c, d, h, H), (U_1, U_2, E) \in G^3$ ;

输出:  $V \in G: (U_1, U_2, E, V)$  被密钥主人认为是一条有效密文。

**注释 15.1** 我们简化了生成有效密文的问题, 即从给定三元组  $(U_1, U_2, E)$  和公钥输出第四个分量  $V$  的问题。将  $V$  看做是一个输入分量, 输出前三个明文分量之一的问题与该问题本质上相同, 但是因为  $V$  不是杂凑函数  $H$  的输入, 所以输出  $V$  是最简单的情况。

**证明** 为了从输入的值构造一条有效的密文, 算法必须输出  $V \in G$ , 满足

$$U_1^{x_1 + y_1 \alpha} U_2^{x_2 + y_2 \alpha} = V \quad (15.3.8)$$

对于输入公钥的拥有者,  $x_1, y_1, x_2, y_2$  是私钥,  $\alpha = H(U_1, U_2, E)$ 。

因为  $G$  是素数  $q$  阶群, 所以  $g_1 \neq 1$  是  $G$  的一个生成元(见推论 5.3)。我们可以记  $r_1 = \log_{g_1} U_1, r_2 = \log_{g_2} U_2, w = \log_{g_1} g_2$ ; 而且存在  $\log_{g_1} c, \log_{g_1} d$ , 并且对于任意的  $V \in G$  存在  $\log_{g_1} V$ 。结合式(15.3.8)和公钥分量  $c, d$  的构造(这在密钥建立过程中被暗含检验过), 我们得到如下线性方程组:

$$\begin{pmatrix} 1 & 0 & w & 0 \\ 0 & 1 & 0 & w \\ r_1 & r_1 \alpha & wr_2 & wr_2 \alpha \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix} = \begin{pmatrix} \log_{g_1} c \\ \log_{g_1} d \\ \log_{g_1} V \end{pmatrix} \pmod{q} \quad (15.3.9)$$

应用高斯消去律, 式(15.3.9)中的矩阵等价于

$$\begin{pmatrix} 1 & 0 & w & 0 \\ 0 & 1 & 0 & w \\ 0 & 0 & w(r_2 - r_1) & w\alpha(r_2 - r_1) \end{pmatrix} \pmod{q} \quad (15.3.10)$$

由于  $(g_1, g_2, U_1, U_2) \notin \mathbf{D}$ , 我们有  $r_1 \neq r_2 \pmod{q}$ ; 同时注意  $w \neq 0 \pmod{q}$  (因为  $g_2 \neq 1$  也是  $G$  的一个生成元)。所以式(15.3.10)是满秩矩阵, 秩为 3, 即三个行向量线性无关。由线性代数的一个简单事实, 对于任意的  $V \in G$ , 方程组(15.3.9)对  $(x_1, y_1, x_2, y_2) \pmod{q}$  有(不惟一的)解。

这样, 我们就证明了对于满足定理条件的输入值,  $G$  中所有的  $q$  个元素都是  $V$  的有效候选, 即对于这些输入值, 每个  $V \in G$  都会使  $(U_1, U_2, E, V)$  成为一条有效密文。但是, 对于密钥的主人, 在这  $q$  个可能中, 只有一个  $V \in G$  满足他/她选择的私钥分量  $(x_1, y_1, x_2, y_2)$  [我们要澄清: 对任何固定的  $V \in G$ , 方程组(15.3.9)的整数解都不惟一, 但是, 很显然任何固定的整数组  $(x_1, y_1, x_2, y_2)$  只能映射到一个  $V \in G$ ]。因此定理中陈述的解决该问题的成功概率成立。□

现在我们完成了对 Cramer-Shoup 体制安全性的证明。

不难发现在该证明中, “归约为矛盾”是一个线性函数:  $A$  攻击该体制的能力被等价地转化成它区分给定四元组是否属于  $\mathbf{D}$  的能力。因此, 我们说对 Cramer-Shoup 体制安全性的归约证明包含一个严谨的归约。

## 15.4 可证明安全的混合密码体制综述

在 8.15 节中我们介绍过混合密码体制, 主要是出于对效率的考虑。混合密码体制也是一

种解决 IND-CCA2 安全而且实际的公钥密码体制方案。本节我们综述一系列的混合加密体制。因为这种方案很多,所以在介绍时不能包括它们的安全性证明;欲知细节,感兴趣的读者可以学习原作。

混合体制输出的密文包含两部分:密钥封装机制(KEM)和数据封装机制(DEM)。这种 KEM-DEM 密文对可以写做:

$$\text{KEM} \parallel \text{DEM} = \mathcal{E}_{pk}^{\text{asym}}(K) \parallel \mathcal{E}_K^{\text{sym}}(\text{Payload\_Message})$$

收到这个密文对后,接收者用他/她的私钥解密 KEM 块,得到短暂对称密钥  $K$ ,然后用  $K$  解密 DEM 块,恢复 Payload\_Message。

如果 KEM 是一个可证明 IND-CCA2 安全的非对称加密方案的输出,那么 DEM 块的 IND 特性就是短暂密钥随机性的自然结果。不难理解 KEM-DEM 结构的混合体制是 IND-CCA2 安全的。事实上,可以把 KEM-DEM 结构下的混合体制看做是得到 IND-CCA2 安全且实际有效的公钥加密的最自然方法。

我们认为混合体制是最自然的方法,是因为它们可以用很低的代价加密任意长度的消息。在应用中,数据的长度可变,大多数情况下,它们的长度都比公钥密码体制中安全参数(例如, RSA-OAEP 中的  $n$  或 Cramer-Shoup 体制中的  $\log_2(\#G)$ )的固定长度要长。因为公钥加密需要的代价比单钥加密高,所以可证明安全的公钥体制,如 RSA-OAEP 和 Cramer-Shoup 体制,在实际中非常可能只是用于构造混合体制中的 KEM 块,而数据的加密则由一系列 DEM 块完成。

混合加密方案包括 Shoup 提出的几个 KEM-DEM 方案[273],Fujisaki 和 Okamoto 提出的 FO 方案[115],Pointcheval 提出的 HD-RSA 方案[234],Abdalla、Bellare 和 Rogaway 提出的 DHAES 方案[4],Shoup 提出的 Cramer-Shoup 体制的一个变形[271],以及 Okamoto 和 Pointcheval 提出的 REACT 方案[225]。

Fujisaki 和 Okamoto 提出的方案可用以下公式表示:

$$\mathcal{E}_{pk}^{\text{hybrid}}(m) = \text{KEM} \parallel \text{DEM} = \mathcal{E}_{pk}^{\text{asym}}(\sigma, H(\sigma, m)) \parallel \mathcal{E}_{G(\sigma)}^{\text{sym}}(m)$$

这里  $G$  和  $H$  都是杂凑函数。在该方案中,KEM 的解密结果是  $\sigma, H(\sigma, m)$ 。接收者用  $\sigma$  作为杂凑函数  $G$  的“种子”,得到单钥加密的密钥  $G(\sigma)$ ;然后用它解密 DEM 块;最后接收者可以通过再计算  $H(\sigma, m)$  验证解密的正确性。所以该方案允许接收者检验密文在途中是否被修改或毁坏。检测密文是否改变是密码体制抗击主动攻击的主要技术。

Pointcheval 的方案 HD-RSA 基于所谓的相关 RSA 困难问题[235]:给定 RSA 的密文  $A = r^e \pmod{N}$ ,求  $B = (r+1)^e \pmod{N}$ 。如果不能求出  $A$  模合数  $N$  的  $e$  次根(RSA 问题),这个问题显然就是难解的。HD-RSA 方案的 KEM 块是简单地对随机数  $r \in \mathbb{Z}_N^*$  计算  $A = r^e \pmod{N}$ 。接收者作为  $N$  的主人当然可以从  $A$  提取出  $r$ ,然后再构造  $B$ 。该方案使用  $K = G(B)$  作为 DEM 块的密钥,这与 Shoup 以及 Fujisaki-Okamoto 的方案一样。

Abdalla、Bellare 和 Rogaway 的方案 DHAES[4]也是在 DEM 块中附加一个消息认证码(MAC, 见 10.3 节)作为数据完整性检验方法的混合体制。两个对称密钥(一个用于 DEM 块中的加密,一个用于 MAC 块中的加密)由一个杂凑函数公式得到:  $H(g^u, g^v)$ ,  $g^u$  是 KEM 块,  $g^v$  是接收者的公钥。显然,公钥  $g^v$  的主人可以在 KEM 块  $g^u$  上用私钥  $v$  得到  $g^w$ ,因此重新构造  $H(g^u, g^w)$ ,从而得到两个对称密钥。没有私钥  $v$ ,解密任务似乎与解计算 Diffie-Hellman 问题(定义 8.1)类似。给定  $g^u, g^v$ ,求  $H(g^u, g^w)$  的问题称为 **hash Diffie-Hellman (HDH) 问题**。



有趣的是,在 DHAES 中,如果  $g^w$  被直接用做加密中的乘数,与 ElGamal 和 Cramer-Shoup 体制相同,那么语义安全性就是基于一个判定问题:判定  $(g, g^u, g^v, g^w (= e/m_b))$  是否是一个 Diffie-Hellman 四元组。在这个混合体制中,杂凑函数的使用使得这个四元组中的四个元素不易得到,所以该判定问题似乎被弱化成一个计算问题。记住我们总是希望用最弱的困难性假设作为安全性的基础。读者可以做一个习题证明 HDH 问题的困难性在 CDH 问题(8.4 节的定义 8.1)和 DDH 问题(13.3.4.3 节的定义 13.1)之间。当然,我们必须注意将困难性假设从 DDH 问题“弱化”到 HDH 问题是有条件的:需要对所使用杂凑函数做某个假设(虽然在我们的简要描述中没有提到)。遗憾的是,这个假设与随机预言机性质非常接近。

Shoup 的混合体制[271]是对 Cramer-Shoup 体制“弱化假设”的变形。在最初的 Cramer-Shoup 体制(算法 15.1)中,消息  $m$  的加密与 ElGamal 相同: $h'm$ 。而在[271]“弱化假设”的变形中, $h'$  在杂凑函数  $H(\dots; h')$  下被隐藏,以防止对 DDH 问题的简单测试。由杂凑值  $H(\dots; h')$  导出对称密钥,用于数据完整性检验机制以及 DEM 加密。Shoup 使用“杂凑规避”来命名他的“弱化假设”变形。

## 15.5 可证明安全的实用公钥密码体制的文献注记

Damgård 最先开始研究可以抗击主动攻击的实际公钥密码体制[89]:在公钥密码体制中加一个数据完整性检验过程挫败主动攻击者。这种方法从此成为设计抗击主动攻击可证明安全密码体制的一般策略。但在 CCA2 模式下, Damgård 的方案(在他的原作中有两个方案)都被证明是不安全的(参考例如[313])。

Zheng 和 Seberry 提出实用的加密和数字签名方案,目标为在 CCA2 模式下是安全的[313, 312]。这些方案中的一般思想是使用杂凑函数加强基于单向函数的教科书公钥方案(基于 ElGamal 的)。这是一个有意义的想法,它后来发展成为可证明安全性的随机预言机模型,这在 15.2 节中我们已经研究过。[312]中 IND-CCA2 模式下的可证明安全性是基于一个非标准的假设,称为“由函数引起的空间惟一可抽样性”(以及计算 Diffie-Hellman 假设)。Soldera 发现 Zheng 和 Seberry 的那些方案中有一个实际上并不是 IND-CCA2 安全的[282, 283]。

在基于 RSA 的随机化填充方案中, Shoup 发现了 RSA-OAEP(见 15.2.3.3 节)安全性证明中的不完全性,紧接着就提出了一个对 OAEP 的改进,称为 OAEP+ [272],其安全性证明中的归约比 Fujisaki 等对 RSA-OAEP 安全性证明中的归约[116]还要严谨。该归约更严谨的原因是 Simon 对 RSA 函数求逆的优势与 Malice 攻破该体制的优势呈线性关系。但是,因为 Simon 对 RSA 函数求逆的时间仍然是允许 Malice 询问随机预言机提问次数的二次函数,所以该归约还不是很有效(回顾 15.2.5 节中 RSA-OAEP 类似的情况)。Boneh 也提出了对 OAEP 的改进,称为简单 OAEP(SAEP)和简单 OAEP+ (SAEP+)[50]。但是这些方法的消息恢复带宽都很低(我们将在 16.4.4.2 节中讨论某些随机化填充方案的消息恢复带宽低的问题)。最近, Coron 等[84]给出了另一个对 RSA 的随机化填充方案,称为消息恢复的概率签名方案(PSS-R,最初由 Bellare 和 Rogaway 提出[27],详细情况见下一章),也可用于加密。本质上,这些作者都极具洞察力地认识到,由于杂凑函数的使用,填充方案中数据完整性检验就不必再引入额外的冗余度,譬如 RSA-OAEP 中的 0 串。我们会在下一章研究这个方案。与 SAEP 和 SAEP+ 相同, PSS-R 用于 RSA 加密时,它的消息恢复带宽也很低(见 16.4.4.2 节)。

与 RSA-OAEP 方案一样,上段提到的随机化填充方案都是在 ROM 下对 IND-CCA2 模式可证明安全的。但是,因为 RSA-OAEP 的 IND-CCA2 可证明安全性又被重新建立(见 15.2.4 节),而且 RSA-OAEP 一直是 RSA 加密标准,更重要的是 OAEP 被证明具有最高的消息恢复带宽,所以我们还不清楚这些新的改进是否可以与作为 RSA 加密标准的 RSA-OAEP 获得类似的契机。

对 OWTP 的填充技术可以获得最有效的方案。但是,OWTP 确实是很少见的函数。RSA 和 Rabin(基于二次剩余)可能是通常公钥密码函数中仅有的两个 OWTP。而且,对填充方案基于随机预言机模型的安全性证明(或论证),人们目前都得不到严谨的“归约为矛盾”。有些研究者考虑对于基于一般单向函数的公钥密码函数,设计归约更严谨的可证明安全方案。还有一些研究者设计的方案由 OWTP 到一般陷门单向函数扩展了填充方案。很多公钥密码函数都不是置换(例如 ElGamal 函数就不是置换),因此这样的扩展非常有用。Fujisaki 和 Okamoto [114]以及 Pointcheval[236]提出了这样的两个推广方案。但是,这些方案都不是最有效的:为了检测错误,解密时需要再加密。因为解密通常是在慢设备上进行的,如智能卡,所以应该避免使用解密繁重的方案。

当然,大量混合加密体制也构成了一类可证明 IND-CCA2 安全的实际公钥密码体制。我们已经在 15.4 节中纵览了关于这类方案的详细文献。

最后,我们要提醒读者,对实用的公钥加密方案,在可证明 IND-CCA2 安全的方案中使用数据完整性检验机制,只是提供了一种安全服务,我们称其为“无源识别的数据完整性”,或“来自 Malice 的完整性”(回顾 10.5 节)。在真实世界的多数应用中,这种安全服务是不充分的。在公钥密码体制中获得源识别的一般方法时使用数字签名。

最近,出现了一个新的公钥密码原型,称为签密。签密方案将加密和签名进行组合一次完成。组合的目的是为获得有效的公钥加密,同时提供额外的对电子商务应用非常重要的安全性服务:消息源识别和不可否认性。由于这种新密码原型的出现(最初由 Zheng[311]在 1997 年提出),在公钥密码体制的可证明安全性概念广泛推广之后,研究者们倾向于将可证明安全性的策略应用到签密方案的设计中。我们在下一章将学习一个可证明安全而且实用的签密方案。

## 15.6 本章小结

本章我们详细描述了两个重要的公钥密码体制,不仅形式化证明了它们具有适于应用的安全性,即在 IND-CCA2 模式下的可证明安全,而且也是实际有效的,其效率类似于相应的教科书式加密。本章的加密方案比先前的逐比特加密方法(例如前一章介绍的方案)又向前跨越了一步,因此是实用的公钥加密方案。

读者可能会认为对于解密过程中包含数据完整性检验的公钥密码体制,发送者使用接收者的公钥对消息进行“数字签名”,而密文就是对这个消息签名的加密。该“签名”方案具有消息恢复特性,因此接收者可以使用他/她的私钥恢复明文并验证该“签名”。这个想法从技术上讲是正确的。我们使用带引号的“数字签名”、“签名”的惟一原因是“签名者”可以是任何人,因此这个密码学变换并没有提供一般意义上的签名;然而,如果不遵循指定的程序或不使用给定的(能有效阻止适应性选择密文攻击的)密钥,伪造“签名”是困难的。这是这样的加密方案(以及本章介绍的两个实用的密码体制)在 CCA2 模式下安全的主要原因。

在这些密码体制的安全性证明过程中,我们还引入并解释了几个重要的概念:安全性证明的随机预言机模型(并指出它的局限性),从“归约为矛盾”得到的形式化证明以及其中归约的严谨性。

我们还给出了各种混合加密方案的一个综述,结合对称加密和非对称加密技术,得到了实用的公钥密码体制。

最后,我们给出了一个文献注记以评述这一课题的发展。

## 习题

- 15.1 RSA-OAEP 中的随机输入有什么作用? 其中的常零串  $0^k$  又有什么作用?
- 15.2 加密算法的带宽是指该算法能够加密密文的长度与安全参数的比值。假设 RSA-OAEP 的一个实例使用 2084 作为安全参数, 160 作为随机输入的长度, 那么这个 RSA-OAEP 实例的带宽是多少?
- 15.3 什么是安全性证明的随机预言机模型?
- 15.4 对于基于随机预言机模型的安全性证明, 它的局限性是什么?
- 15.5 15.2.1 节中 RO 的仿真为什么必须从一个初始化为空的有序表来构造?
- 15.6 对于安全性基于计算复杂性问题的密码体制, 它的安全性证明中“归约为矛盾”中的“矛盾”指的是什么?
- 15.7 归约证明中的询问密文为什么必须是随机的?
- 15.8 在 RSA-OAEP 的安全性证明中, Simon 为什么必须使用攻击者两次以上(包括两次)?
- 15.9 虽然 1024 比特的模数可以抵抗现在的分解技术, 那么对于 RSA-OAEP, 为什么还认为这样大小的模数太小了?
- 15.10 Cramer-Shoup 体制也使用一个杂凑函数。那么该体制的安全性证明需要该函数具有随机预言机特性吗?
- 15.11 假设 Cramer-Shoup 体制被修改成按如下方式加密:

$$e \leftarrow h' + m$$

(因此解密执行减法), 其他部分均保持不变。证明修改后的方案是 CCA2 安全的, 即任意主动攻击都可以被检测到。它是 IND-CCA2 安全的吗?

- 15.12 为什么计算  $g^*h'(\bmod p)$  的代价可以只度量为一个模指数运算的代价?
- 15.13 将算法 15.2 扩展到  $f^*g^*h'(\bmod p)$  的情况。
- 15.14 什么是混合密码体制?
- 15.15 在实际应用中, 机密数据的大小一般都比公钥密码体制的安全参数大得多, 而单钥加密的密钥又比该安全参数小得多。为了安全地传输这样的数据, 你会选择下面的哪一个算法? (i) RSA, (ii) AES, (iii) RSA-OAEP, (iv) ElGamal, (v) Cramer-Shoup, (vi) 混合密码体制。

## 第 16 章 强可证明安全的数字签名方案

### 16.1 引言

在数字签名方案的定义(10.4 节的定义 10.2)中,我们约定“压倒性”概率  $\text{Verify}_{pk}(m, s) = \text{False}$  if  $(m, s)$  是一个没有使用指定签名过程而伪造的消息-签名对;但对于第 10 章介绍的签名方案,我们还没有研究过这个“压倒性”概率应该有多大。正如我们在 10.4.9 节中讨论过的,数字签名的教科书式安全性定义即“从零开始”伪造签名的困难性,非常弱以至于不适用于应用。因此,对于第 10 章中的签名方案,如果说我们进行了安全性证明的话,那么这些证明远远不是形式化的,以至于不能提供足够的可信度,并且也太弱了以至于不适用于应用。在第 10 章我们只考虑非形式化的弱安全性证明,其真正的原因是那时我们还缺少技术准备,尚不能进行形式化的强安全性证明。

在前两章中,我们研究了证明公钥密码体制安全性的形式化方法,包括强安全性定义(如 IND-CCA2)、“归约为矛盾”的方法、安全性证明的随机预言机模型和标准模型。现在我们做好了技术准备,可以进一步研究分析数字签名方案安全性的形式化方法了。与对公钥密码体制加强安全性分析时的情况类似,在我们研究对数字签名方案的加强安全性分析时将涉及以下两个问题:

- **依赖攻击数字签名方案最一般方法的签名伪造的困难性** 对数字签名最一般的攻击是适应性选择消息攻击。适应性攻击者拥有目标用户的公钥,并且可以将该用户用做预言机签名服务的提供者(含义在 8.2 节中给出),对它选择的任何消息签名。然后它根据收集到的消息-签名对,适应性改写它的提问。我们可以把这种攻击方式看成是攻击者从目标签名者那里获得伪造签名的训练课程。在提问了足够多的适应性选择消息并得到相应的签名之后,即在足够的训练之后,这个攻击者的目标是输出一个新的消息-签名对,它对于目标用户的公钥是有效的。这里的“新”意味着用户在此之前从未对该消息签名。
- **利用形式化证据建立的安全性论据** 使用“归约为矛盾”的证明方式证明安全性。对安全签名方案的这种归约是一个有效的变换,它表明任何成功的伪造算法(如在适应性攻击下)都可以用做解决计算复杂性理论中某个著名难题的“黑盒”。这里“矛盾”是因为人们普遍相信不存在有效的算法可以解决这个著名难题。

Goldwasser、Micali 和 Rivest 在他们的开创性工作[129]中对数字签名的这两个问题进行了系统的考察。他们还实现了一个可抗击(存在性的)适应性选择消息攻击的签名方案。这个方案使用了“无爪”(Claw-free)置换对的概念:非正式地讲,它们是普通定义域上的两个置换  $f_0$  和  $f_1$ ,对这两个置换来说,要找一个三元组  $(x, y, z)$  满足  $f_0(x) = f_1(y) = z$  在计算上是不可行的。Goldwasser 等用整数分解问题实现了他们的“无爪”置换对(详见[129])。这个签名方案有一个优点,它可以签名任意的随机串而不在该串上添加任何可识别的冗余,如消息格式化不使用杂

凑函数。但是,这个方案以逐比特的方式签名消息,因此人们认为它对于应用还不是很理想。尽管如此,Goldwasser 等工作[129]构成了数字签名方案强(即适于应用)安全性概念的重要基础。

### 16.1.1 本章纲要

16.2 节介绍数字签名方案最强的安全性概念。16.3 节对 ElGamal 族签名方案进行形式化归约的安全性证明。16.4 节介绍基于随机化填充技术和陷门单向置换(主要是 RSA 和 Rabin 函数)的适于应用的签名方案。16.5 节研究签密方案及其适于应用的安全性。

## 16.2 数字签名的强安全性定义

这里我们介绍本章需要用到的定义。

首先,一个数字签名方案可以记为  $(\text{Gen}, \text{Sign}, \text{Verify})$ , 定义 10.2(见 10.4 节)给出了它们的定义。我们在第 10 章已经看到,数字签名中一般都使用密码学杂凑函数(使用的目的大都是为了防止存在性伪造),因此在本章中我们要考虑签名方案中的  $\text{Sign}$  和  $\text{Verify}$  使用一个或更多强杂凑函数的情况。强杂凑函数的意思是,当我们证明一个签名方案的安全性时,会形式化地将该方案中使用的杂凑函数模型化,使其具有 10.3.1.2 节中描述的随机预言机特性。事实上,本章将要提供的所有安全性论证都是在证明安全性的随机预言机模型下给出的(回顾 15.2.1 节)。

对使用杂凑函数的数字签名方案进行适应性选择消息攻击的游戏,现在我们就给出一种渐进定义。

**定义 16.1 适应性选择消息攻击** 假设  $k$  是一个正整数,对签名方案  $(\text{Gen}, \text{Sign}, \text{Verify})$  的一个适应性伪造者是一个(关于  $k$  的)(概率)多项式时间算法。它的输入是公钥  $pk$ , 其中  $(pk, sk) \leftarrow_U \text{Gen}(1^k)$ , 它试图伪造对于  $pk$  的签名。允许伪造者要求并获得它所选择的消息的签名。这一点是通过允许伪造者(关于  $k$  的)多项式次询问签名和杂凑算法来模型化的。称这个伪造者为  $(t(k), \text{Adv}(k))$  攻破该签名方案,如果它在时间  $t(k)$  内,输出一个有效签名的概率是  $\text{Adv}(k)$ , 即输出一个消息-签名对  $(m, s)$ , 满足  $\text{Verify}_{pk} = \text{True}$ ,  $m$  是一条关于该方案中所使用的杂凑函数的可以识别消息,而且不是在这之前签名者已经输入给  $\text{Sign}$  的消息。这里  $t(k)$  是关于  $k$  的一个多项式,  $\text{Adv}(k)$  是关于  $k$  的一个不可忽略量。

对于不可忽略量(函数)的概念,可以回顾 4.6 节。

我们简化了该定义,因为没有给出关于签名和杂凑函数提问这两个次数表达式的陈述。省略的这两个表达式都是关于  $k$  的多项式:因为伪造者是(关于  $k$  的)多项式时间的算法,所以它最多就只能进行(关于  $k$  的)多项式次的签名和杂凑函数提问。

**定义 16.2 安全的签名方案** 称一个签名方案  $(\text{Gen}, \text{Sign}, \text{Verify})$  是  $(t(k), \text{Adv}(k))$  安全的,如果对于所有足够大的  $k$ , 不存在可以  $(t(k), \text{Adv}(k))$  攻破该方案的伪造者。

我们将以归约为矛盾的方式使用定义 16.2。假设一个给定签名方案是  $(t(k), \text{Adv}(k))$  可攻破,  $t(k)$  是关于  $k$  的一个多项式,  $\text{Adv}(k)$  是关于  $k$  的一个不可忽略量。我们可以构造一个



归约变换,将  $t(k)$  转换为  $t'(k)$ ,将  $Adv(k)$  转换为  $Adv'(k)$ ,使得一个基本的困难问题成为  $(t'(k), Adv'(k))$  可解的。如果该归约足够有效,那么  $t'(k)$  就是足够小的,  $Adv'(k)$  也足够接近  $Adv(k)$ ,因此也足以达到不可忽略。因为普遍认为该基本困难问题不是  $(t'(k), Adv'(k))$  可攻破的,所以我们得到了一个矛盾,完成证明。关于有效归约的含义以及归约应该尽可能有效性的重要性,读者可以回顾 15.2.5 节。

与我们在前一章研究过的对公钥密码体制的“归约为矛盾”方法类似,本章证明签名方案的归约也由一个特殊的代理 Simon 仿真器来完成。Simon 通过对伪造者选择的进行签名,在他与伪造者的交互中起到了目标签名者的作用。这通过仿真签名预言机来完成。为了使伪造者完全发挥它伪造签名的能力,仿真签名预言机必须与真正的签名机不可区分。因为伪造者是多项式有界的,所以我们使用定义 4.15(见 4.7 节)中的多项式时间不可区分性概念就足够了。

本章其余部分将伪造者命名为 Malice,它是一个主动攻击者。

## 16.3 ElGamal 族签名的强可证明安全

ElGamal 签名方案(见 10.4.6 节)以及这一类的签名(如 10.4.8.1 节中 Schnorr 的方案和 10.4.8.2 节的 DSS)出现以后,在很长的时间(1985 ~ 1996)内,人们广泛相信伪造这种签名的困难性应该多少与解决有限域中大子群上的离散对数问题有关。但直到 1996 年才建立形式化证据(形式化证明)。

为了将在 ElGamal 签名族方案中伪造签名的困难性同计算离散对数的困难性联系起来,Pointcheval 和 Stern 成功地给出了肯定证据[237]。他们通过使用一个强大的工具——安全性证明的随机预言机模型(ROM)[23]——做到了这一点。读者可以回顾 15.2.1 节来温习使用 ROM 进行安全性证明的一般概念(那里,基于 ROM 的证明是证明公钥加密体制的安全性)。对基于 ROM 证明 ElGamal 签名族方案安全性的一般方法,Pointcheval 和 Stern 基于 ROM 的证明是一个颇具代表性的实例。

### 16.3.1 三元组 ElGamal 族签名

现在我们介绍 ElGamal 族签名方案的一种典型形式,它在 ROM 下是可证明不可伪造的。这种形式的方案以签名密钥  $sk$ 、公钥  $pk$  和一个比特串消息  $M$  为输入,输出  $M$  的签名:三元组  $(r, e, s)$ 。其中

- $r$  称为承诺;它承诺一个短暂整数  $\ell$ ,我们称为承诺对象,它独立于之前签名中用过的所有承诺对象;构造承诺的一般形式是  $r = g^\ell \pmod{p}$ ,这里的  $g$  和  $p$  是该签名方案公开参数的一部分;
- $e = H(M, r)$ ,  $H()$  是一个密码学杂凑函数;
- $s$  称为签名;它是承诺  $r$ 、承诺对象  $\ell$ 、消息  $M$ 、杂凑函数  $H()$  和秘密签名密钥  $sk$  的一个线性函数。

我们把这样的签名方案称为三元组签名方案。



算法 10.3 中给出的最初 ElGamal 签名方案不是三元组签名方案,因为它没有使用杂凑函数,因此也不能抗存在性伪造(更没有考虑适应性选择消息攻击)。但是,使用了杂凑函数并因此抗存在性伪造的方案,(如 10.4.7.2 节中描述的一个变形)是三元组形式的。

Schnorr 签名方案(见算法 10.4)也是三元组形式的。由 Schnorr 签名方案的签名算法产生的对消息  $M$  的签名是  $(r, e, s)$ , 其中  $e = H(M, r)$ ,  $H()$  是一个杂凑函数;但是在 Schnorr 签名中,没有必要将  $r$  发送给验证者,因为该值可以由  $g^s y^e$  计算得到。

现在我们要介绍 Pointcheval 和 Stern 证明三元组签名方案不可伪造性的归约方法,这个方法被称为分叉归约技术。

### 16.3.2 分叉归约技术

我们在 10.4.7.1 节中证明了如果 ElGamal 族中的签名方案不是一次使用一个短暂密钥(承诺对象  $l$  或者承诺  $r$ ),就会导致签名私钥的泄露。签名私钥的泄露可以有效地解决一个困难问题:求模一个大素数的群中元素(公钥)的离散对数。

对三元组 ElGamal 族签名方案的归约安全性证明利用了承诺重放技术揭示签名私钥。对这样的签名方案,可以把一个成功伪造者归约为一个代价类似的签名私钥提取器。因为后者是求模大素数群中元素(公钥)的离散对数,它是公认的困难问题(见 8.4 节的假设 8.2),所以所谓的成功伪造签名会有类似的难度,而这两个问题难度的类似程度取决于归约的有效性。

在三元组 ElGamal 签名方案的基于 ROM 的归约安全性证明中,杂凑函数被理想化为一个随机函数,称为“随机预言机”(RO),其特性已在 10.3.1.2 节中描述过。在 ROM 下,所有的 RO 都由 Simon 仿真器仿真。另外,Simon 还要仿真签名过程并回答 Malice 的签名提问。这样,Simon 就可以为 Malice 提供允许他得到的必要训练课程了,这是为了让 Malice 在他的伪造签名任务中准备充分。如果 Malice 确实是一个成功的伪造者,那么他就可以从该训练课程中受益,从而以不可忽略的概率输出一个伪造的消息-签名对。Simon 将利用这个伪造的签名解决一个困难问题,在三元组 ElGamal 签名的情况下,该问题是有限域中的离散对数问题。图 16.1 给出了 Simon 利用 Malice 解决一个困难问题的归约技术。

在接下来的两小节中,我们尽量直观地描述 Pointcheval 和 Stern 的归约技术。因此,尽管我们采用与 Pointcheval 和 Stern 相同的逻辑推理,但我们的概率估计结果和他们给出的公式不完全相同。针对归约的严谨性来说,我们的结果是 Pointcheval 和 Stern 结果的一个上界。尽管如此,我们的这个上界足以对大的安全参数产生一个有合理意义的矛盾。对此有进一步研究兴趣的读者可以参阅 [238] 研究 Pointcheval 和 Stern 更复杂的概率估计。

#### 16.3.2.1 非适应性攻击下的不可伪造性

首先,让我们考虑三元组 ElGamal 签名方案在非适应性攻击下的不可伪造性。

假设  $(\text{Gen}(1^k), \text{Sign}, \text{Verify})$  是三元组形式 ElGamal 签名方案的一个实例(即算法 10.3 的三元组形式),该方案中的素数  $p$  满足:存在一个整除  $p-1$  的  $k$  比特素数  $q$ , 而且  $(p-1)/q$  没有大素数因子。

假设 Malice 是对  $(\text{Gen}(1^k), \text{Sign}, \text{Verify})$  的一个成功伪造者。Simon 仿真器屏蔽了 Malice 的所有通信信道,如图 16.1。但是,在非适应性攻击背景下,因为 Malice 未曾要求得到一个签名,所以 Malice 和 Simon 之间的交互没有“仿真签名训练”。

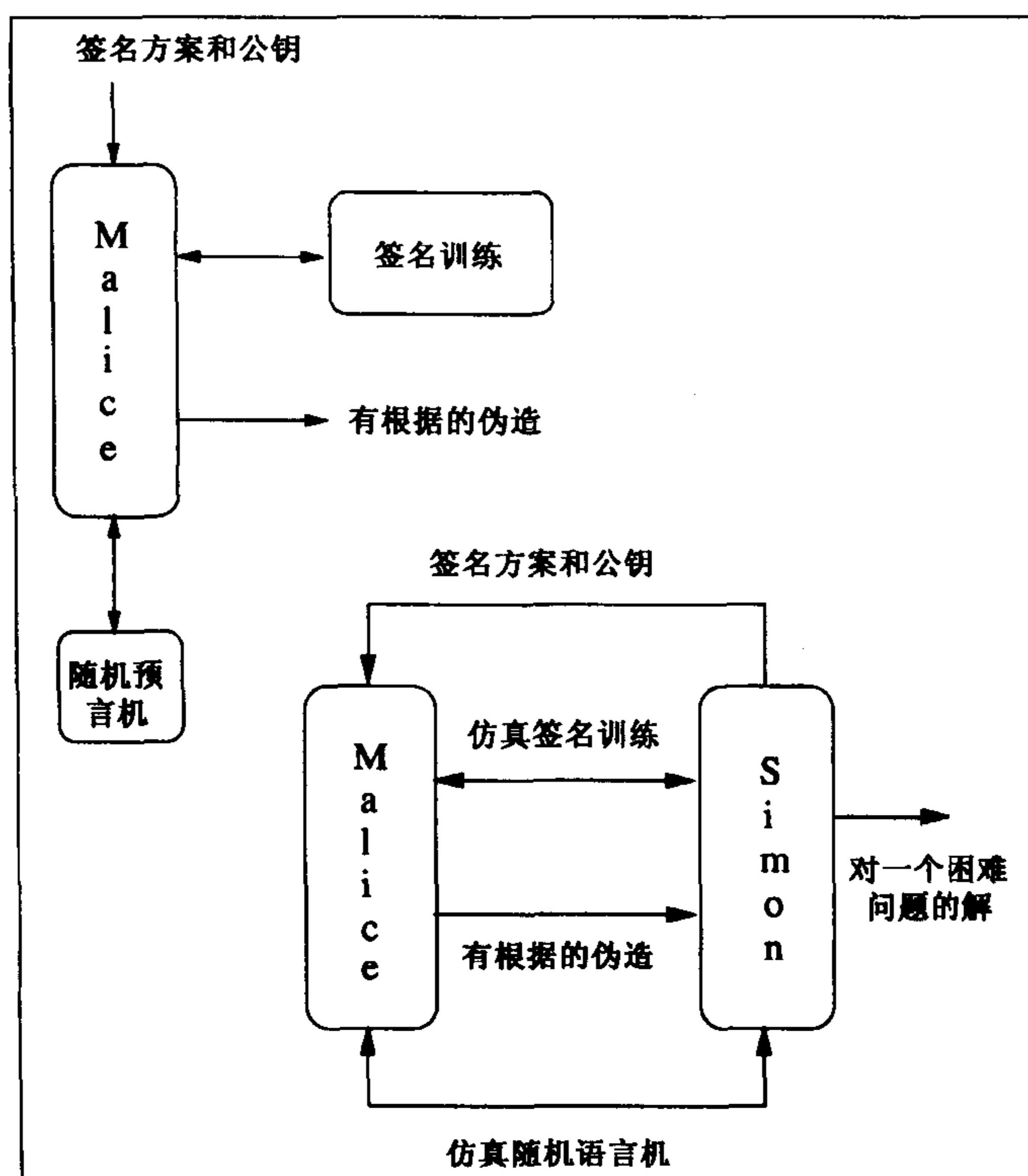


图 16.1 从伪造签名到解决困难问题的归约

Simon 选择一个随机元素  $y \in \mathbb{Z}_p^*$ 。他的目标是得到以模  $p$  生成元  $g$  为基底的  $y$  的离散对数,即求满足  $y \equiv g^x \pmod{p}$  的  $x$ 。Simon 将以下述方式把 Malice 当做一个黑盒: Malice 对一条选择消息成功伪造的新签名会给 Simon 提供足够的信息求解离散对数。我们希望读者已经直觉地意识到了输入问题(即  $y$ )必须是任意的,否则,这个归约就不是一个有用算法。

假设 Malice 伪造签名的成功概率是  $Adv(k)$ ,  $Adv(k)$  是关于  $k$  的一个不可忽略量,他伪造签名所需的时间是  $t(k)$ ,  $t(k)$  是关于  $k$  的一个多项式。我们要求出 Simon 求离散对数的成功概率  $Adv'(k)$  以及需要的时间  $t'(k)$ 。当然,我们要将  $(t'(k), Adv'(k))$  和  $(t(k), Adv(k))$  联系起来。

### Malice 的第一批运行

Simon 运行 Malice  $1/Adv(k)$  次。因为 Malice 是一个成功的伪造者,所以只要满足一个条件(很快就会给出),他就会以概率 1(因为他被运行了  $1/Adv(k)$  次)输出消息  $M$  在方案 (Gen, Sign, Verify) 下的一个有效签名。即

$$e = H(M, r)$$

$$y'r^s \equiv g^e \pmod{p}$$

其中  $|e| = k$ 。

Simon 必须满足 Malice 的条件是允许后者得到一定数量的 RO 函数  $H$  的计算。在 ROM 下,如图 16.1, Malice 必须向 Simon 做 RO 提问。Simon 通过对 RO 的仿真来回答询问:他通过维护有序元素  $((M_i, r_i), e_i)$  (如以  $M_i$  排序)组成的一个  $H$  表仿真  $H$ , 其中  $(M_i, r_i)$  是提问,  $e_i$  是随机回答。

因为 Malice 是多项式有界的,所以他只能进行  $n = q_H$  次 RO 提问,  $q_H$  是(关于  $k$  的)多项式有界的。假设

$$Q_1 = (M_1, r_1), Q_2 = (M_2, r_2), \dots, Q_n = (M_n, r_n) \quad (16.3.1)$$

是 Malice 的  $n$  个不同 RO 提问。假设

$$R_1 = e_1, R_2 = e_2, \dots, R_n = e_n$$

是 Simon 对这些询问的  $n$  个回答。因为  $|H| = 2^k$ , 所以 Simon 的回答在集合  $\{1, 2, 3, \dots, 2^k\}$  中均匀分布。

由于 Simon 的回答是均匀分布的,所以当 Malice 输出一个对  $M$  的有效伪造签名  $(r, e, s)$  时,他一定提问过  $(M, r)$ ,并得到了回答  $e = H(M, r)$ 。也就是说,一定存在某个  $i \in [1, n]$  有  $(M, r) = (M_i, r_i)$ 。 $(M, r)$  没有被提问过的概率是  $2^{-k}$  (即 Malice 在没有提问 Simon 的情况下正确地猜测到了 Simon 的均匀随机回答  $R_i = e_i$ )。因为  $2^{-k}$  是可忽略的,所以我们知道  $((M, r), e)$  在 Simon 的  $H$  表中。

我们再次指出,必须记住的重要一点:如果没有向 Simon 做过 RO 提问,或没有使用 Simon 的回答, Malice 就成功不了,除了一个非常小的可忽略概率  $2^{-k}$ 。观察到这一点,我们就可以想像:似乎是 Malice“被迫”对式(16.3.1)中  $n$  个消息中的某一个伪造签名。

#### Malice 为成功分叉的第二批运行

Malice 在同样条件下再运行另外的  $1/\text{Adv}(k)$  次。也就是说,他要做与式(16.3.1)中完全相同的  $n$  次提问。但是,这次 Simon 要重排他的  $n$  个均匀随机回答。

我们必须注意,由于重排的回答仍然在集合  $\{1, 2, 3, \dots, 2^k\}$  中均匀分布,所以这些回答还是正确的(关于这一点将在注释 16.1 中进一步解释)。

得到第二组  $n$  个正确回答后, Malice 必须再次完全发挥他的伪造能力,并以概率 1 输出一个对  $M'$  的新伪造签名  $(r', e', s')$ 。与对 Malice 第一组运行的讨论一样,  $(M', r')$  必须是式(16.3.1)中对于某个  $j \in [1, n]$  的  $Q_j$ , 除去一个很小的概率  $2^{-k}$ 。

在 Malice 的两组运行中,当两个伪造的消息-签名对  $(M, (r, e, s))$  和  $(M', (r', e', s'))$  满足  $(M, r) = (M', r')$  时,“成功分叉 Malice 的 RO 提问”这一事件就会发生,如图 16.2。注意,在 Malice 的每一组运行中,他可以伪造一个对  $(M_i, r_i)$  的签名,  $i \in_U [1, n]$  是均匀随机的,没有必要固定。由生日悖论(见 3.6 节),我们知道这个事件发生的概率大约是  $1/\sqrt{n}$ 。注意,这与在 Malice 第二组运行中的固定  $i$  的情况不同,那会导致(在固定点)成功分叉的概率为  $1/n$ 。

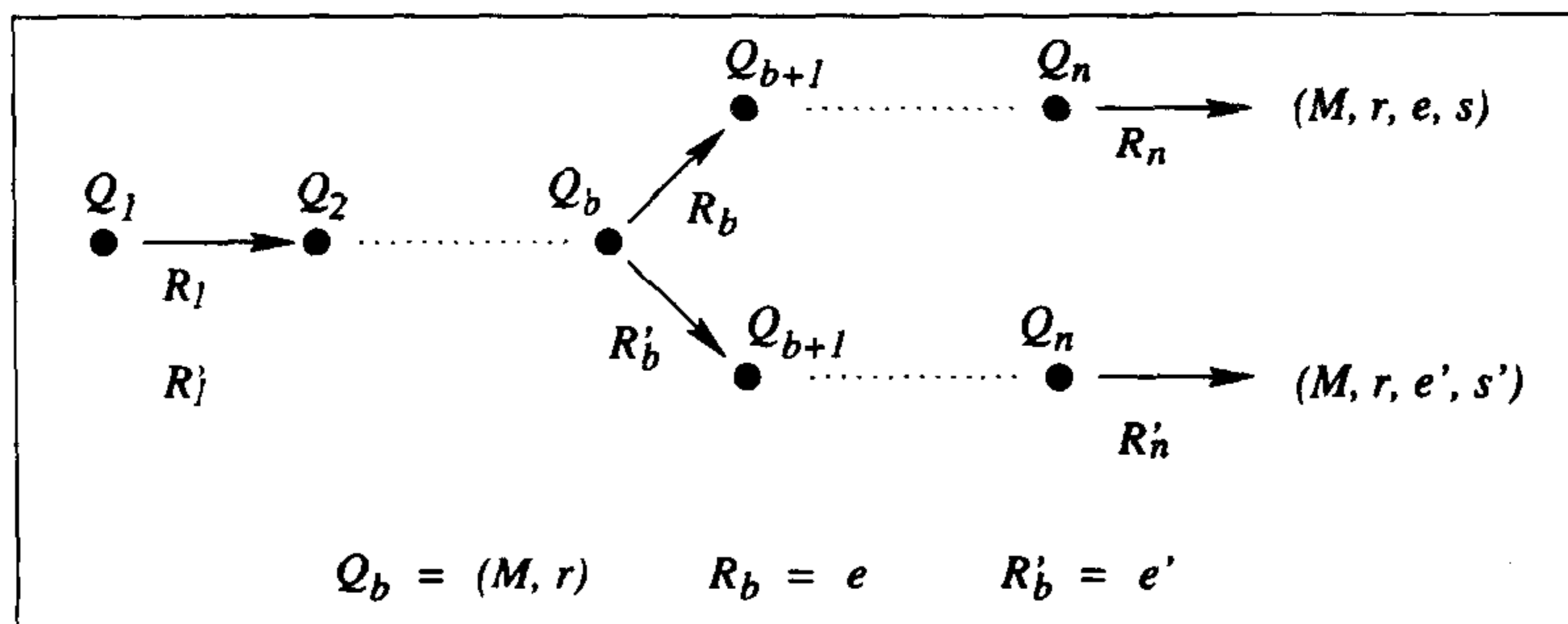


图 16.2 对随机预言提问的成功分叉回答

因为  $n$  是多项式有界的, 所以  $1/\sqrt{n}$  是一个不可忽略的量。也就是说, 以不可忽略的概率  $1/\sqrt{n}$ , Simon 得到两个有效的伪造  $(r, e, s)$  和  $(r', e', s')$ 。进一步注意到在第二组运行中 Simon 均匀随机地重排了他的回答, 因此必然以“压倒性”概率  $1 - 2^{-k}$  有  $e' \neq e \pmod{q}$ 。

由一个成功的分叉, Simon 可以求目标离散对数值。下面我们看看如何做到这一点。

### 求离散对数

由这两个有效的伪造签名, Simon 可以计算

$$y'r^s = g^e \pmod{p}$$

$$y'r^{s'} = g^{e'} \pmod{p}$$

因为  $g$  是模  $p$  的生成元, 所以存在某个整数  $\ell < p - 1$ , 满足  $r = g^\ell \pmod{p}$ 。同时注意到  $y = g^x \pmod{p}$ , 我们有

$$xr + \ell s = e \pmod{p}$$

$$xr + \ell s' = e' \pmod{p}$$

由  $e' \neq e \pmod{q}$ , 推出  $s' \neq s \pmod{q}$ , 所以有

$$\ell \frac{e - e'}{s - s'} \pmod{q}$$

最后, 如果  $q \mid r$ , 那么归约失败。这个条件使我们可以对 ElGamal 签名方案实施 Bleichenbacher 攻击[42], 这在 10.4.7.1 节的第一个提醒中提到过。但是, Bleichenbacher 攻击是通过恶意选择公钥参数做到的, 所以对于随机选择的公钥, 事件  $q \mid r$  发生的概率显然是  $1/q$ , 这样我们就不必在意 Malice 可能对某个整数  $\xi$  成功伪造签名  $(M, \xi q, H(M, \xi q), s)$ , 因为这样的成功伪造只是有效签名中可忽略的一部分。由此, 因为以“压倒性”概率有  $r$  与  $q$  互素, 所以 Simon 可以求得

$$x = \frac{e - \ell s}{r} \pmod{q}$$

记住,  $(p - 1)/q$  没有大素数因子, 所以可以很容易地进一步求得  $x \pmod{p - 1}$ 。

因为  $r, e, e'$  都在 Simon 的两个 RO 表中,  $s, s'$  是 Malice 的输出, 所以 Simon 确实可以使用上述方法求得以  $g$  为底  $y$  模  $p$  的离散对数。在这个方法中, Simon 将 Malice 当做一个黑盒: 他不关心也不研究 Malice 是如何工作的; 只要 Malice 工作, Simon 就可以工作。

### 归约结果

至此我们得到了如下归约结果:

i) Simon 求得离散对数的优势是

$$Adv'(k) \approx \frac{1}{\sqrt{q_H}}$$

因为  $q_H$  是(关于  $k$  的)多项式有界的, 所以  $Adv'(k)$  是关于  $k$  的不可忽略量。

ii) Simon 所需要的时间是

$$t' \approx \frac{2(t + q_H)}{Adv(k)}$$

其中  $t$  是 Malice 伪造一个签名所需要的时间。我们将在 16.3.2.3 节中讨论该归约算法的有效性。

这种基于 ROM 归约证明的理论基础称为分叉引理[237]。

**注释 16.1** 分叉归约技术可行的原因是 Simon 仿真器对 RO 回答的重排,这使得 Malice 的一组询问得到了两组完全独立的回答。似乎 Malice 很愚蠢,没有觉察对同一组询问的回答不同。其实不然,Malice 作为一个成功的伪造者还是很聪明的。我们可以认为 Malice 是一个概率算法,只要该算法是在正确的环境下,并且得到了正确分布的 RO 回答,他惟一的功能就是输出有效的伪造签名。我们不能认为这个概率算法可能有某个额外功能,譬如该算法可能有和人类一样的意识,从而发觉通信环境中是否有人在进行欺骗。事实上,Simon 给了 Malice 正确分布的回答,所以根本没有欺骗他。□

### 16.3.2.2 适应性选择消息攻击下的不可伪造性

现在我们考虑在适应性选择消息攻击下的不可伪造性。

这种情况下的归约技术本质上与非适应性选择密文攻击中的相同。但是,现在 Malice 除了可以进行 RO 提问,还可以进行签名提问( $q_s$  次)。因此,Simon 仿真器除了要回答 RO 提问,还要回答签名提问,该回答必须能通过 Malice 使用  $\text{Verify}_{pk}$  的验证。虽然没有签名私钥,但是 Simon 必须做到这一点。事实上,签名正是他企图通过 Malice 的帮助才能得到的! Simon 的签名过程也是通过仿真完成的。

因此,我们只需要证明在 ROM 下,Simon 确实可以高质量地满足 Malice 的签名提问。

因为签名算法使用一个由 RO 模型化的杂凑函数,所以在 ROM 下对每个签名提问  $M$ , Simon 都选择一个新的  $r < p$ ,代替 Malice 作一个 RO 提问  $(M, r)$ ,然后将 RO 的回答和签名回答都返回给 Malice。对每一次签名询问,Simon 生成的新  $r$  都完全服从签名过程;Simon 不再使用任何已经使用过的  $r$ 。

下面就是 Simon 需要做的事情。为了签名一个提问  $M$ ,Simon 选择小于  $p-1$  的随机整数  $u$  和  $v$ ,令

$$\begin{aligned} r &\leftarrow g^u y^v \pmod{p} \\ s &\leftarrow -rv^{-1} \pmod{p-1} \\ e &\leftarrow -ruw^{-1} \pmod{p-1} \end{aligned}$$

Simon 返回  $e$  作为 RO 提问  $(M, r)$  的回答,返回  $(r, e, s)$  作为  $M$  的签名(即作为对签名提问  $M$  的回答)。读者可以验证返回的签名确实是有效的。事实上,这个仿真签名的算法正是在 10.4.7.2 节中我们给出的一个存在性伪造,在那里我们验证过了这种存在性伪造的有效性。

在 ROM 下,这个仿真签名的分布与用 RO 代替杂凑函数  $H$  的签名算法的签名分布相同。这就是 Malice 不会发现任何异常的原因。所以,Simon 提供的“仿真签名训练”(如图 16.1)是高质量的,由此 Malice 不仅会对 RO 应答很满意,也会对这些签名应答满意。他的伪造能力能够完全发挥,使用与 16.3.2.1 节中相同的归约就会正如所愿地导致一个矛盾。

证明完成。定理 16.1 总结了我们得到的安全结论。

**定理 16.1** 假设  $(\text{Gen}(1^k), \text{Sign}, \text{Verify})$  是三元组 ElGamal 族签名方案的一个实例, 素数  $p$  满足: 存在一个整除  $p-1$  的  $k$  比特素数  $q$ , 且  $(p-1)/q$  没有大素数因子。如果一个适应性选择消息的伪造者可以在时间  $t(k)$  内以  $\text{Adv}(k)$  的优势攻破该方案, 那么可以在时间  $t'(k)$  内以  $\text{Adv}'(k)$  的优势求解模  $p$  的离散对数问题, 其中

$$t'(k) \approx \frac{2 \cdot (t(k) + q_H \cdot \tau) + O_B(q_s \cdot k^3)}{\text{Adv}(k)}$$

$$\text{Adv}'(k) \approx \frac{1}{\sqrt{q_H}}$$

$q_s$  和  $q_H$  分别是签名数和  $H$  预言提问次数,  $\tau$  是回答一个  $H$  提问所需要的时间。□

在这个结论中,  $k^3$  是模一个  $k$  比特整数的指数运算所需要的比特操作数(我们在 4.3.2.6 节中推出了模指数运算的这个立方时间复杂度表达式)。

### 16.3.2.3 讨论

- 我们又一次看到安全性证明中 ROM 的功效。基于 ROM 对三元组 ElGamal 族签名方案的安全性证明揭示了这样一个事实: 如果签名算法确实是随机函数, 那么伪造签名的最简单方法是首先求解离散对数, 然后像真正的签名者那样签名。这与我们在第 9 章中给出的关于比特安全性的研究结果一致。

所以, 基于 ROM 的证明表明, 对于实际中使用杂凑函数而不是 RO 的签名方案, 进行攻击的最薄弱环节可能是该方案中使用的杂凑函数, 除非攻击者认为攻击该杂凑函数比求解离散对数问题更困难。因此我们认为, 基于 ROM 的安全性证明方法的重要性在于它间接地提示我们在细心设计时应当特别注意的地方。

- 我们看到 Simon 求解离散对数问题的优势是  $\frac{1}{\sqrt{q_H}}$ , 这里  $q_H$  是 Malice 可以向  $H$  随机预言提问的次数。为了使 Simon 获得求解离散对数问题的一个常数优势, 该归约要运行  $\sqrt{q_H}$  次。这将使 Simon 的运行时间增长为

$$\sqrt{q_H} \cdot \frac{2 \cdot (t + q_H \cdot \tau) + O_B(q_s \cdot (\log p)^3)}{\text{Adv}}$$

如果我们认为杂凑函数可以有效地计算, 那么允许专业的伪造者计算  $2^{50}$  次(与 15.2.5 节中我们的实例相同)杂凑函数就是合理的。因此, 在归约证明中, 我们必须允许 Malice 做  $2^{50}$  次 RO 提问, 即  $q_H = 2^{50}$  是一个合理的假设。在这个合理的假设下, 考虑 Simon 时间代价的主要部分  $q_H^{3/2}$ , 我们得到 Simon 求解离散对数问题的时间是

$$O\left(\frac{2^{75}}{\text{Adv}}\right)$$

这个时间代价表明我们的归约不是非常有效。对于  $p$  是 1024 比特的素数, 导致的矛盾不是很有意义, 尤其当  $\text{Adv}$  很小时。而当  $p$  是 2048 比特的素数时, 这个归约可以说是有意义的。

虽然这个归约不具备理想的效率, 但是 Pointcheval 和 Stern 基于 ROM 的分叉归约方法给出了对三元组 ElGamal 族签名方案的第一个归约安全性证明。



- 颇具讽刺意味的是,我们之所以能够给出适应性选择消息攻击的不可伪造性(数字签名最强的安全性概念)证明,原因正是该签名方案本身固有的可存在性伪造这一缺陷。但是,这一点与 15.2.4 节中证明 RSA-OAEP 方案的安全性时“Shoup 的最初尝试”情况不同,那里 Shoup 建议使用 3 作为 RSA 加密的公开指数。对于基于陷门单向函数的数字签名方案,本身固有的存在性伪造这一“缺陷”并不是一种本质缺陷(而是一种性质),而使用公开指数为 3 的 RSA 加密却是真正的缺陷。
- 虽然数字签名标准(DSS,见 10.4.8.2 节)不是三元组签名方案(杂凑函数的输入只是消息比特串,而不是消息和承诺值),但是在 ROM 下证明 DSS 具有同等程度的不可伪造性没有什么本质上的技术难度。如果我们假设,对于给定的密钥,Simon 可以存储整个历史中所有提问过的 RO 提问和签名提问的消息,那么该形式化证明就能很顺利。因为在这种方式下,对已有的消息能得到对应的已有回答。基于 ROM 对三元组 ElGamal 签名方案的成功证明或许表明应该把 DSS 修改成三元组的形式,即也应该对承诺值求杂凑值。
- Pointcheval 和 Stern[237]还给出了对 Fiat 和 Shamir[110]签名的安全性证明,这是因为 Fiat 和 Shamir 的签名方案本质上是一个三元组签名。这个签名方案是通过修改后面一章要介绍的一个零知识识别方案而得到的。

### 16.3.3 重行归约方法

还有另外一种证明三元组 ElGamal 族签名方案不可伪造性的归约方法。这种方法称为**重行**,是由 Feige、Fiat 和 Shamir[107]为证明 Fiat 和 Shamir 的零知识识别方案[110]的正确性(我们将在 18.2.2 节研究零知识协议的正确性)而创造的。因为识别协议可以很容易地转化为 Fiat 和 Shamir 的三元组签名方案(虽然不在 ElGamal 族中),所以重行方法自然也可用于三元组 ElGamal 族签名方案。这一事实最后在[224]中得到了证明。现在我们简要介绍证明三元组 ElGamal 族签名方案安全性的重行归约方法。

在重行归约方法中,我们仍然假设 Malice 伪造签名的优势是  $Adv$ 。Simon 将运行 Malice 多次,次数与  $1/Adv$  成正比(正好是  $3/Adv$  次)。

我们设想一个巨大的二元矩阵  $H$ ,它有  $q$  行  $q$  列。其中  $q$  个行对应三元组 ElGamal 签名方案中第一个元素所有可能的随机选择, $q$  个列对应这个方案中第二个元素所有可能的随机选择。 $H$  中的分量  $h_{i,j}$  是 1,如果  $(i,j,s)$  是一个有效的签名;否则是 0。称一个行是重行,如果它包含至少两个 1。

对于这个矩阵,一个非常简单但至关重要的事实是:

**引理 16.1 重行引理** 1 既在  $H$  中又在重行中的概率至少是  $1/2$ 。

这只是因为重行中的 1 比其他行中的多。

因为 Malice 是对三元组签名方案的一个成功伪造者,他具有的优势为  $Adv$ ,我们知道  $H$  中有  $Adv \cdot q^2$  个 1。运行  $1/Adv$  次 Malice,就一定会输出一个正确的伪造  $(i,j,s)$ 。由重行引理, $i$  是重行的概率至少是  $1/2$ 。现在再运行  $2/Adv$  次 Malice,保持承诺  $i$  不变,Malice 会成功地伪造另一个有效签名  $(i,j',s'), j' \neq j$ 。

我们已经知道有了这两个伪造的签名就可以成功求得所需要的离散对数值,也就是导致一个所希望的矛盾。

在对重行技术的描述中,我们集中解释了其思想的直观性,所以我们没有介绍应用生日悖论效应时的结果,这会使得引理中的概率更大。有关使用生日悖论效应的重行归约方法,对它精确系统的描述,读者可以参阅[224]。

## 16.4 适于应用的 RSA 和 Rabin 签名方法

RSA 和 Rabin 函数是陷门单向置换(OWTP,关于为什么以及怎样使用一种可借鉴的方式把 Rabin 函数转变成 OWTP,请回顾 14.3.6.1 节),所以基于这些函数的教科书式签名方案(10.4.2 节中的教科书 RSA 签名和 10.4.4 节中的教科书 Rabin 签名方案)都是确定性算法。这就意味着对于给定的密钥对  $(sk, pk)$  和消息  $M$ ,这些签名算法输出的对  $M$  的签名是由  $(sk, pk)$  和  $M$  惟一确定的。

在密码学中,确定性是我们不期望的性质。对于教科书式 Rabin 签名方案,确定性也是我们在 10.4.5 节中能对该方案进行有效攻击的原因:适应性选择消息攻击使得 Malice 可以得到同一选择消息的两个不同的平方根,从而分解模数。因此,RSA 和 Rabin 签名方案的适于应用的形式必须是概率的。

### 16.4.1 具有随机化填充的签名

Bellare 和 Rogaway 首先研究了使用 RSA 和 Rabin 函数概率签名的方法[27]。他们把这个方法命名为概率签名方案(PSS)。这是 RSA(和 Rabin)函数的一个基于随机化填充技术的方案。为了简化表述,下面我们只考虑 RSA 的情况。

与 OAEP 填充方案(见图 15.1)类似,PSS 填充方案也是用杂凑函数构造的,并且本质上也和 OAEP 方案的思想相同。在 RSA-OAEP 加密方案中,加密过程是一个使用 RSA 函数单向性的变换,而在 RSA-PSS 签名方案中,签名过程是一个使用 RSA 函数陷门性的变换,因为签名者拥有私钥。

现在让我们具体描述 RSA-PSS 方案,一个重要的适于应用的数字签名方案。

### 16.4.2 概率签名方案

我们仅以 RSA 为例说明算法,对 Rabin 的情况类似。

图 16.3 给出了 PSS 填充的图示。该签名方案在算法 16.1 中具体描述。

签名算法和验证算法均使用了两个杂凑函数。第一个是  $H$ ,称为压缩函数, $H: \{0,1\}^* \mapsto \{0,1\}^{k_1}$ ,第二个是  $G$ ,称为生成函数, $G: \{0,1\}^{k_1} \mapsto \{0,1\}^{k-k_1-1}$ 。在分析安全性时,这些杂凑函数都被模型化为 RO。

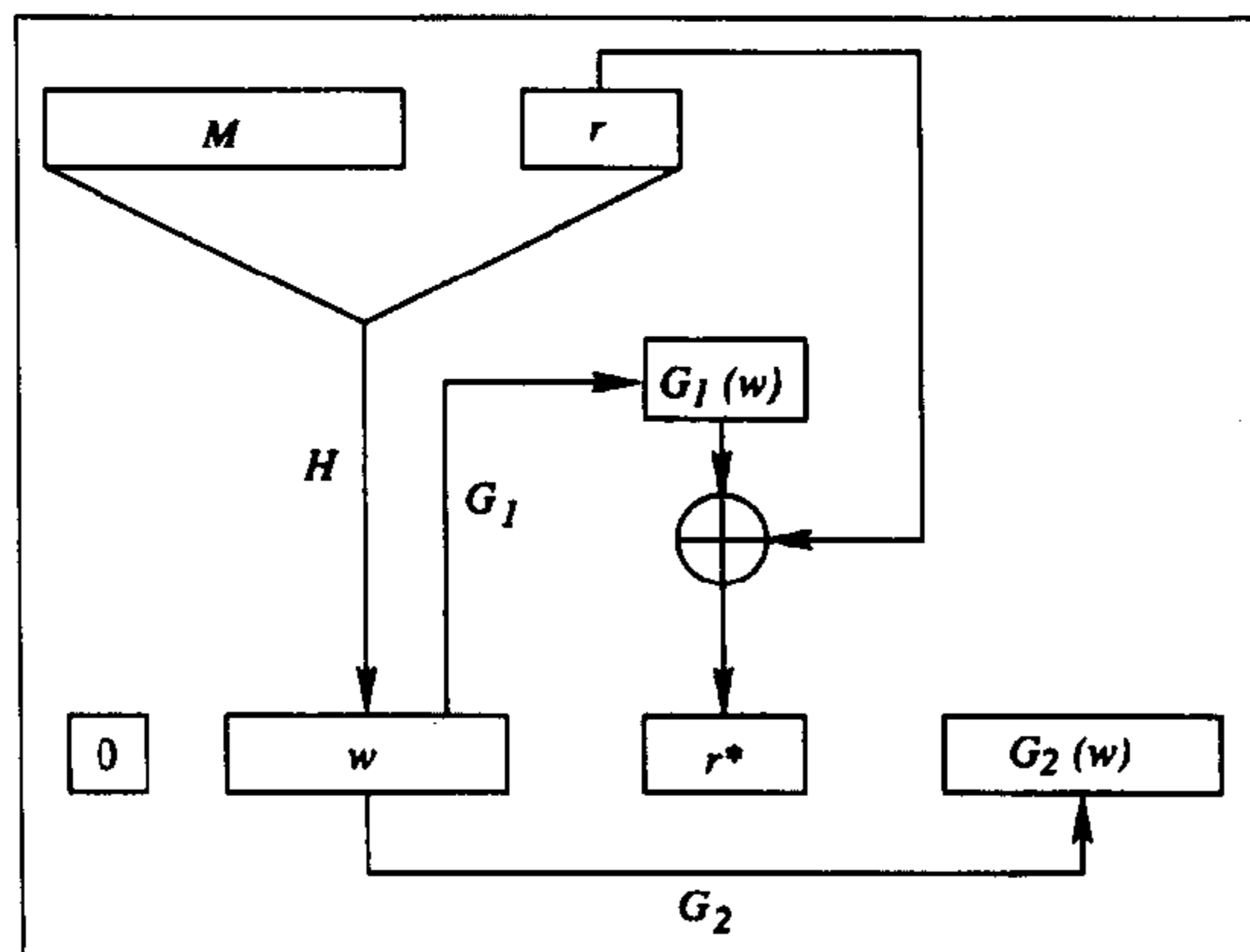


图 16.3 PSS 填充

最开始的 0 有什么作用? 由杂凑函数以及随机输入的长度, 我们知道该填充结果有  $k-1$  比特。所以, 在填充结果前加一个 0 才是  $k$  比特串, 而且当我们将这个串看成整数时, 它比  $N$  小。为了模指数运算的正确, 这是必需的。另一种保留 1 比特带宽并同时保证填充结果比  $N$  小的方法是填充结果恰好是  $k$  比特, 但这需要签名者执行试错检验。我们在算法 10.6 对 RSA-OAEP 填充的具体描述中已经介绍过这种方法, 该算法是对最初算法[25]的一个很小的修正。

#### 16.4.2.1 安全性证明

对 RSA-PSS 方案签名不可伪造性的形式化证明可以使用基于 ROM 的归约技术给出(参阅[27])。这个形式化证明也是由“归约为矛盾”得到的: 一个成功的伪造将导致 RSA 函数的求逆, 而这是一个公认的困难问题。在证明用于加密的 RSA 填充算法的安全性时, 我们也给出了一个归约的构造(即我们在 15.2 节中研究过的证明 RSA-OAEP 安全性的归约构造), RSA-PSS 安全性证明的归约构造同它非常类似。

#### 算法 16.1 概率签名方案(PSS)

##### 密钥参数

假设  $(N, e, d, G, H, k_0, k_1) \leftarrow_{\mathcal{U}} \text{Gen}(1^k)$ , 其中  $(N, e, d)$  是 RSA 密钥材料,  $(N, e)$  是公钥,  $d = e^{-1} \pmod{\phi(N)}$  是私钥;  $k = |N| = k_0 + k_1$ ,  $2^{-k_0}$  和  $2^{-k_1}$  是可忽略的量;  $G, H$  是杂凑函数, 满足:

$$G: \{0,1\}^{k_1} \mapsto \{0,1\}^{k-k_1-1}, H: \{0,1\}^* \mapsto \{0,1\}^{k_1}$$

(\*  $G$  输出的比特串被分成两个子串, 一个记为  $G_1$ , 它包括前面最重要的  $k_0$  比特, 另一个记为  $G_2$ , 它包括其余  $k - k_1 - k_0 - 1$  比特 \*)

##### 签名生成

$\text{SignPSS}(M, d, N) = r \leftarrow_{\mathcal{U}} \{0,1\}^{k_0}; w \leftarrow H(M \parallel r); r^* \leftarrow G_1(w) \oplus r;$   
 $y \leftarrow 0 \parallel w \parallel r^* \parallel G_2(w);$   
 return( $y^d \pmod{N}$ )。

##### 签名验证

$\text{Verify}(M, U, e, N) = y \leftarrow U^e \pmod{N};$   
 Parse  $y$  as  $b \parallel w \parallel r^* \parallel \gamma;$   
 (\* 也就是说, 令  $b$  是  $y$  的第一个比特,  $w$  是后续的  $k_1$  比特,  $r^*$  是再接下来的  $k_0$  比特,  $\gamma$  是其余比特 \*)  
 $r \leftarrow r^* \oplus G_1(w);$   
 if( $H(M \parallel r) = w \wedge G_2(w) = \gamma \wedge b = 0$ ) return(True)  
 else return(False)。

特别地, RSA-PSS 安全性证明的归约也是将一个成功的签名伪造变换成一个对 RSA 函数的部分求逆, 和 15.2.3.4 节中的对 RSA-OAEP 安全性的归约证明相同(那里, 一个成功的 IND-CCA2 攻击导致了  $s^*$  的泄漏, 而  $s^*$  是询问密文  $c^*$  的一个部分  $e$  次根)。但是, 对签名的证明

比对加密的证明简单:不用像加密的情况那样,再运行 Malice,对 RSA 函数的部分求逆就可以直接导致完全求逆。这归因于签名伪造的计算本质:在一个成功的签名伪造中, Malice 必须向 Simon 提供一个消息-签名对,可以用单向函数(这里是 RSA 函数)验证它。相反,在成功的 IND-CCA2 攻击中, Malice 向 Simon 提供的消息少得多,只是一比特的猜测,所以 Simon 得不到任何能把猜测明文和询问密文联系起来的单向函数。这样得到的求逆只是部分求逆。因此,在加密的情况下,为了得到该函数的完全求逆,归约必须改变这些部分逆的位置,借助 Malice 的再次运行。

在对 RSA-PSS 签名方案的安全性证明中,一次就能完全求逆的直接结果是得到一个有效归约: Malice 伪造签名的优势  $Adv$  严谨地转换为 Simon 的优势,即  $Adv' \approx Adv$ 。Bellare 和 Rogaway 称这个严谨的归约结果为基于填充 RSA 签名方案的精确安全性。

由于对 RSA-PSS 签名方案的安全性证明与对 RSA-OAEP 加密方案的安全性证明在概念上相似,而且陈述该归约的细节很繁琐,这里我们就不描述这个归约了。感兴趣的读者可以参阅 [27] 中的详细情况。

### 16.4.3 PSS-R: 消息可恢复的签名

从 RSA-OAEP 加密方案允许私钥主人恢复被加密的消息这个事实,我们可以考虑相反方向的一个问题:基于填充的消息可恢复签名方案也可以允许任何人恢复被签名的消息,只要这个人有正确的公钥。这正是 RSA-PSS-R 方案,即消息可恢复概率签名方案所做的。Bellare 和 Rogaway 对于 RSA 和 Rabin 给出了 PSS-R 填充方案 [27]。

我们要介绍与最初 PSS-R 填充方案稍有不同的一个变形方案。这种变形是 Coron 等人提出的 [84]。之所以选择介绍这种变形是因为文 [84] 的作者证明了他们的变形不仅在使用 RSA 函数的陷门性生成签名时,签名是安全的,而且在使用 RSA 函数的单向性生成密文时,加密也是安全的。这里签名的安全性是指在适应性选择消息攻击下的不可伪造性,而加密的安全性是指在 IND-CCA2 下的安全性。

### 16.4.4 签名和加密通用的 PSS-R 填充

图 16.4 给出了 PSS-R 填充的两个图示;一个是 Bellare 和 Rogaway 最初的方案 [27],另一个是 Coron 等的变形方案 [84]。签名和加密通用的填充技术在算法 16.2 中具体描述。

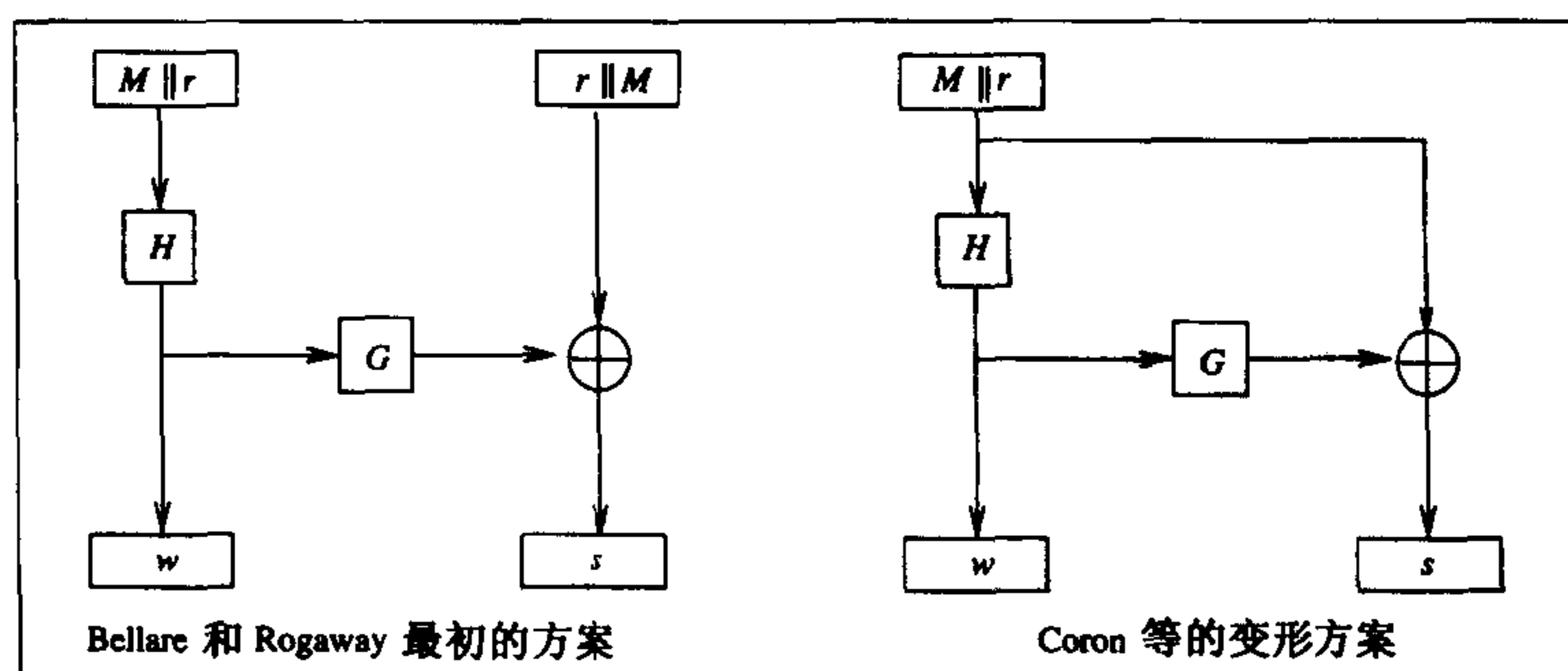


图 16.4 PSS-R 填充

在这个通用的 RSA 填充方案中,签名和加密过程称为 PSS-R 填充。其输入为一条消息  $M \in \{0,1\}^{k-k_1-k_0}$ 、一个 RSA 指数和一个 RSA 模数;RSA 指数是  $d$ ,用于生成签名,而  $e$  则用于加密。注意,与消息不限长的 PSS 签名方案不同,现在消息必须限长,长度为  $k-k_1-k_0$ 。签名验证和密文完整性验证的解密过程称为 PSS-R 去填充。其输入为一个数  $U < N$  和 RSA 密钥材料,输出是  $\{\text{True}, \text{False}\} \cup \{0,1\}^{k-k_1-k_0}$ ;当输出的第一部分为 True 时,输出的其余比特是恢复的消息;否则,输出的其余比特是一个无意义的串 Null。

#### 算法 16.2 签名和加密通用的 RSA 填充方案

##### 密钥参数

假设  $(N, e, d, G, H, k_0, k_1) \leftarrow \text{Gen}(1^k)$ , 其中  $(N, e, d)$  是 RSA 密钥材料,  $(N, e)$  是公钥,  $d = e^{-1}(\text{mod } \phi(N))$  是私钥;  $k = |N| = k_0 + k_1$ ,  $2^{-k_0}$  和  $2^{-k_1}$  是可忽略的量;  $G, H$  是杂凑函数, 满足:

$$G: \{0,1\}^{k_1} \mapsto \{0,1\}^{k-k_1-1}, H: \{0,1\}^{k-k_1-1} \mapsto \{0,1\}^{k_1}$$

##### 签名生成或消息加密

PSS-R 填充  $(M, x, N) =$

1.  $r \leftarrow \{0,1\}^{k_0}; w \leftarrow H(M \parallel r); s \leftarrow G(w) \oplus (M \parallel r); y \leftarrow (w \parallel s);$
2. if  $(y \geq N)$  go to 1;
3. return  $(y^x \text{ mod } N)$ 。

##### 签名验证或含密文证实的解密

PSS-R 去填充  $(U, x, N) = y \leftarrow U^x \text{ mod } N;$

Parse  $y$  as  $w \parallel s;$

(\* 如今  $w$  是第一个  $k_1$  比特,  $s$  是其余的  $k-k_1$  比特 \*)

Parse  $G(w) \oplus s$  as  $M \parallel r;$

(\* 如今  $M$  是第一个  $k-k_1-k_0$  比特,  $r$  是其余的  $k_0$  比特 \*)

if  $(H(M \parallel r) = w)$  return  $(\text{True} \parallel M)$

else return  $(\text{False} \parallel \text{Null})$ 。

#### 16.4.4.1 安全性证明

对 RSA-PSS-R 加密和签名方案安全性的证明在概念上与 (i) RSA-OAEP 加密的情况和 (ii) RSA-PSS 签名的情况相同。再次因为概念的相似性以及描述细节相当繁琐,我们在这里也不给出这个归约。详细情况读者可以参阅[84]。

#### 16.4.4.2 讨论

- 在 PSS-R 填充中,为了确保填充结果作为一个整数小于  $N$ ,我们用了试错检验。该检验重复  $i$  次的概率是  $2^{-i}$ 。另一种方法,PSS 填充方案中使用的头部加 0 技术也可以用在里。

- 当 PSS-R 填充用于加密时,密文的完整性检验是通过检验杂凑值完成的。这个方法与 OAEP 填充方案有所不同,后者检查一个 0 串作为恢复的冗余。
- 对 PSS-R 填充方案基于 ROM 的 IND-CCA2 安全性分析在本质上与我们对 RSA-OAEP 给出的相同:通过归约到 RSA 函数的一个部分逆而完成,其中  $w$  没有泄漏,即如果 Malice 以优势成功攻破该方案,那么在与仿真器 Simon 进行的攻击游戏中, Malice 必定以类似于 Adv 的优势已经提问过随机预言机  $G$ 。因为攻击者的一次运行只能得到部分逆,所以为了获得足够的信息对这个函数完全求逆,这个归约至少需要攻击者的两次运行。正如我们在 15.2.4 节中看到的,为了使归约导致一个有意义的矛盾,它运行 Malice 的次数不能超过两次(以使归约成为 2 次多项式)。

即使是 Malice 运行最少的两次,该归约也已经远远不是严谨的了。读者可以回顾 15.2.5 节中关于归约不严谨的后果。为了获得一个有意义的矛盾,不严谨的归约要求 RSA-PSS-R 加密方案的 RSA 模数必须至少为 2048 比特。

- 运行 Malice 的最少次数(两次)要求该填充技术满足  $|w| > \frac{|N|}{2}$ ,从而有  $|M| \parallel r| \leq \frac{|N|}{2}$ 。所以,用于加密的 RSA-PSS-R 填充方案恢复消息的带宽太低:恢复的消息长度必须小于模数长度的一半。对于典型的参数设置  $k = |N| = 2048$ ,  $k_0 = 160$ ,我们可以得到最大的  $|M| = \frac{|N|}{2} - k_0 = 1024 - 160 = 862$ ,即  $|M|$  最多只能达到  $|N|$  的 42%。
- 和我们在 16.4.2.1 节中讨论过的 RSA-PSS 签名一样,对 RSA-PSS-R 基于 ROM 的安全性证明(对适应性选择消息攻击的不可伪造性)也有一个严谨的归约。这是因为一个成功的签名伪造可以导致一次运行就能够对 RSA 函数完全求逆。所以,与前一段对加密安全性证明的讨论不同,对签名的安全性证明不需要条件  $|w| > \frac{|N|}{2}$ 。我们认为  $k_0, k_1$  的大小只要满足  $2^{-k_0}, 2^{-k_1}$ ,对猜测攻击可忽略,  $k_0 = k_1 = 160$  就足以满足这一点。所以,  $|M| = k - k_0 - k_1$  可以很大。以典型的情况  $k = |N| = 2048$ ,  $k_0 = k_1 = 160$  为例,可以得到  $|M| = 2048 - 320 = 1728$ ,即  $|M|$  可以达到  $|N|$  的 84%。

## 16.5 签密

为了避免伪造信的内容,并确保其保密性,一般的实用做法是作者需要先对该信签名,然后将其封在一个信封里,交给邮递员。安全通信中的这种普通业务应用了数字签名和数据加密技术,而且通常都是分开应用:发送方先对一个消息签名,然后加密该结果;接收方先解密该密文,然后验证签名。

签名和加密使机器循环操作,而且还引入了消息扩展。对消息进行密码学操作的代价通常是用消息扩展率和接发双方所需的计算时间度量。对于直接先签名后加密的过程,以既认证又保密的方式传递一条消息,其代价必然是数字签名和加密的代价之和。通常情况下,这并不是完成该任务的一种经济方式。

签密是以一种很有效的方式获得数字签名和加密组合功能的公钥密码原型。因此它能提供三种经常使用的安全性服务:保密性、认证性和不可否认性。因为人们经常都同时需要这些服务,所以 Zheng 提出了签密[311],以比直接组合数字签名和加密方案更经济的方式提供这些安全性。



### 16.5.1 Zheng 的签密方案

Zheng 提出了两个非常类似的签密方案,分别称为 SCS1 和 SCS2[3011]。这两个方案又分别应用到两个非常类似的 ElGamal 族签名方案中,分别称为 SDSS1 和 SDSS2。

回忆 16.3.1 节,在三元组 ElGamal 签名  $(r, e, s)$  中,承诺  $r$  通常由  $r = g^k \pmod{p}$  计算,其中  $g$  和  $p$  都是公钥材料,承诺对象  $k$  是独立于先前签名用过的所有值的整数。进一步回忆,在作为三元组 ElGamal 方案的 Schnorr 签名方案(算法 10.4)中,签名者没有必要将承诺发送给接收者,因为签名生成的方式允许接收者通过计算  $r = g^s y^e \pmod{p}$  恢复该承诺。

因此,如果消息发送者(作为消息的签名者)以一种特殊的方式计算该承诺,使得只有预期的接收者可以恢复(如使用该接收者的公钥计算),那么该承诺值就可以用做发送者和接收者之间共享的对称密钥(种子),从而应用对称加密提供消息的保密性。

这或多或少就是 Zheng 签密方案要做的:使用三元组 ElGamal 族签名方案的可恢复承诺值作为对称密钥对消息进行对称加密,三元组方案的签名就是所要的签名。从这个简要和抽象的描述中,我们已经看出可以将一个签密的消息写做三元组  $(c, e, s)$ ,  $c$  是对称加密算法输出的密文,  $(e, s)$  是三元组签名的第二个和第三个分量;而三元组签名的第一个分量(传统上记为  $r$ )只有消息接收者可以恢复。

由于 SCS1 和 SCS2 非常相似,所以我们只介绍 SCS1。我们在算法 16.3 中给出了它的描述。为了简化说明,我们的描述使用传统的三元组 ElGamal 签名方案术语,除了我们使用  $K$  替代  $r$ (三元组 ElGamal 签名方案中的承诺)来表示该值被用做对称密钥以外。

现在我们证明算法 16.3 中描述的系统既是一个加密体制又是一个签名方案,即 (i) Bob 的解密过程确实返回 Alice 签密过的同一明文消息, (ii) Alice 对该消息签了名。

为了证明(i),只要证明 Bob 可以恢复 Alice 加密过的  $K = g_B^u \pmod{p}$  即可。Bob 的恢复过程是

$$K = (g^e y_A)^{x_B} \equiv (g^{x_B})^e (g^{x_A x_B})^s \equiv y_B^{s(e+x_A)} \equiv y_B^u \pmod{p}$$

#### 算法 16.3 Zheng 的签密方案 SCS1

##### 系统参数建立

一个可信的权威机构执行下列步骤:

1. 建立系统参数  $(p, q, g, H)$ ;  
(\* 这些参数与 Schnorr 签名方案(算法 10.4)的相同 \*)
2. 另外,建立对称加密算法  $\mathcal{E}$ ;  
(\* 例如, AES 就是  $\mathcal{E}$  的一个很好候选 \*)

为了系统范围内用户的使用,公开参数  $(p, q, g, H, \mathcal{E})$ 。

##### 参与者公私钥的建立

用户 Alice 选择一个随机数  $x_A \in_U \mathbb{Z}_q$ , 计算

$$y_A \leftarrow g^{x_A} \pmod{p}$$

Alice 的公钥材料是  $(p, q, g, y_A, H, \mathcal{E})$ ; 她的私钥是  $x_A$ 。

**签密**

为了将  $M$  的签密发送给 Bob, Alice 执行:

1. 从  $[1, q]$  中随机选择一个  $u$ , 计算  $K \leftarrow y_B^u \pmod{p}$ , 将  $K$  分成合适长度的  $K_1$  和  $K_2$ ;
2.  $e \leftarrow H(K_2, M)$ ;
3.  $s \leftarrow u(e + x_A)^{-1} \pmod{q}$ ;
4.  $c \leftarrow \mathcal{E}_{K_1}(M)$ ;
5. 将签密密文  $(c, e, s)$  发送给 Bob。

**解签密**

收到 Alice 发送的签密密文  $(c, e, s)$  后, Bob 执行:

1. 从  $e, s, g, p, y_A$  和  $x_B$  恢复  $K: K \leftarrow (g^e y_A)^{x_B} \pmod{p}$ ;
2. 将  $K$  分成  $K_1$  和  $K_2$ ;
3.  $M \leftarrow \mathcal{D}_{K_1}(c)$ ;
4. 只有当  $e = H(K_2, M)$  时, 接受  $M$  作为 Alice 发送的有效消息。

所以, Bob 确实能恢复 Alice 加密的  $K$ 。使用从  $K$  中分离出来的  $K_1$ , Bob 当然可以解密密文  $c$  并恢复消息  $M$ 。

对于(ii), 我们注意到当恢复  $K = g^u \pmod{p}$  后,  $(K_2, e, s)$  构成了对恢复消息的三元组 ElGamal 签名。所以算法 16.3 中的系统确实是一个签名方案。

**16.5.1.1 讨论**

- **效率** SCS1 方案在计算以及通信带宽方面都非常有效。在计算方面, 为了签密, 发送者执行一个模指数运算、一个杂凑运算和一个对称加密运算; 如果我们把指数表达式  $(g^e y_A)^{x_B}$  改写为  $g^{ex_B} y_A^{x_B}$ , 并且使用算法 15.2 进行计算, 那么接收者解签密的计算量与发送者类似。在通信带宽方面, 因为消息的对称加密不会引起任何数据扩展, 所以签密密文可以用  $2|q|$  比特再加上被签密消息的比特数进行发送。这和用 ElGamal 族签名传递一个签名(与被签消息一起)的带宽相同。而且, 对称加密算法的使用使得该方案适用于有效发送大批量的数据(例如, 使用 CBC 操作模式的分组密码, 见 7.8.2 节)。在本质上, SCS1 可以看做是一个混合公钥加密方案, 关于这种方案在 15.4 节中已有概述。
- **安全性** 对于签名的不可伪造性, Zheng 给出了该方案的一个合理证明。因为我们看到 SCS1 方案本质上是一个承诺可恢复的三元组 ElGamal 签名, 所以签名在适应性选择消息攻击下的不可伪造性可以直接由对 Pointcheval 和 Stern 提出的三元组 ElGamal 签名方案[237]基于 ROM 的证明得到(我们在 16.3 节中学习过这个技术)。但是, 对于消息的保密性, 因为涉及到对称加密算法, Zheng 没有给出 IND-CCA2 安全性的归约证明。构造 IND-CCA2 安全性的归约证明有一个很大的障碍, 这可能是由于这样一个原因: 在适应性选择密文攻击下, 只有目标接收者可以恢复承诺值  $K$ 。
- **不可否认性** 不可否认性, 即参与者不能否认该消息的来源, 这对于很多应用都是一种重要的安全服务, 如电子商务。数字签名能提供这种服务的原因是消息的签名是普遍

可验证的:当双方因为一个消息-签名对发生争执时,可以让第三方仲裁。在签密的情况下,如果签名不能做成普遍可验证的,那么不可否认性服务就要丧失。Zheng 的签密方案就是这样。因为(三元组)签名的验证需要恢复承诺值  $K$ ,而恢复又必须使用接收者的私钥。所以第三方就不能直接仲裁。Zheng 建议一旦接收者(Bob)和发送者(Alice)发生争执,Bob 就与仲裁者进行一次零知识证明,证明他拥有 Alice 的签名。该方案中没有给出任何零知识证明协议。虽然设计这样的协议并不困难,但是将一个简单的验证过程转化成一个交互式协议很令人烦恼。这是 Zheng 签密方案最严重的缺陷。

### 16.5.2 一箭双雕:采用 RSA 签密

Malone-Lee 和 Mao 提出了一种签密方案,称为“一箭双雕”(TBOS)[184](稍后会解释这个名词)。TBOS 签密方案用 RSA 实现。他们给出了对于消息保密性和签名不可伪造性的强安全性的归约证明。这两个证明虽然基于 ROM,但都在 RSA 函数求逆困难的假设下。

TBOS 签密方案非常简单,而且确实也可以很简单地描述。它使用 RSA 签名和加密函数对消息进行“双层包裹”:发送者(如 Alice)首先将消息“包裹”在她自己的 RSA 函数的陷门部分,得到消息的一个签名,然后再将签名“包裹”在预期接收者(如 Bob)的 RSA 函数的单向部分,得到该签名的加密。所以,如果我们记 Alice 的 RSA 公钥材料和私钥材料分别为  $(N_A, e_A)$ ,  $(N_A, d_A)$ , Bob 的分别为  $(N_B, e_B)$ ,  $(N_B, d_B)$ ,那么 TBOS 签密就是消息  $M$  的如下“双层包裹”:

$$[M^{d_A}(\bmod N_A)]^{d_B}(\bmod N_B)$$

虽然这个想法在概念上很简单,但是对于教科书式 RSA,这种方式的“双层包裹”一般都不起作用。这是因为 Alice 的 RSA 模数可能比 Bob 的大,所以“里层包裹”结果作为一个整数可能已经比“外层包裹”使用的模数大。

尽管如此,我们已看到适于应用的 RSA 方案,不管是加密还是签名,只有在消息用随机化填充方案处理过之后才“包裹”消息。对于这样的 RSA 方案,收发双方应该商定一个填充和去填充方案,因此系统范围内的用户必须使用同样长度的模数。

系统范围内的用户使用了相同长度的模数,“双层包裹”就能很好地工作。如果“内层包裹”结果超过了“外层包裹”的模数,那么发送者只要从“内层包裹”中“砍掉”一个比特(如最重要的比特)就可以了。“砍掉”一个比特后,剩余的整数必然比“外层包裹”的模数小(很快就会说明这一点),因此可以直接“包裹”。记住,这种 RSA 密文的接收者必须进行密文完整性检验,该步骤允许接收者使用试错检验将“砍掉”的比特找回来。这就是整个思想。

现在,对于  $|N_A| = |N_B| = k$ ,假设  $\text{Padding}(M, r) \in \{0, 1\}^k$  表示对消息  $M$  用随机输入  $r$  进行随机化填充。那么 Alice 发送给 Bob 的消息  $M$  在 TBOS 签密方案下看起来似乎是如下“双层包裹”:

$$[\text{Padding}(M, r)^{d_A}(\bmod N_A)]^{d_B}(\bmod N_B)$$

从对 TBOS 签密方案的抽象描述中,我们可以看到这个方案的三个良好性质:

- 它生成了紧凑的密文:签密密文和没有签名的 RSA 密文大小相同,或者说和没有加密的 RSA 签名的大小相同。这就是将该方案命名为“一箭双雕”的原因。这个性质在很多电子商务应用中都非常吸引人,在这些应用中,一条短消息(如授权支付的信用卡卡号)需

要在 Internet 上发送,并且既要提供秘密性保护又要提供授权支付的不可否认性。在这些应用中,TBOS 可以提供一个短密码。这不仅可以获得效率,而且有利于减少电子商务协议的工程复杂度。

- 它以一种非常直接的方式提供了不可否认性:接收者 Bob 在“解开”签密密文后,或者可能是在修复被“砍掉”的比特之后,他就拥有了发送者 Alice 通常形式的 RSA 签名:  $\text{Padding}(M, r)^{d_A} \pmod{N_A}$ 。任何第三方都可以用通常的方式验证该签名。
- 对 TBOS 方案的安全性证明可以沿用其他适于应用的 RSA 填充方案的证明,并以归约的方式给出。虽然这些证明基于 ROM,但是它们只依赖于一个著名的困难问题(RSA 问题和假设,8.7 节中的定义 8.4 和假设 8.3),这是我们非常渴望的。

现在我们说明 Bob 总能正确地解签密。如果  $N_A < N_B$ ,这显然为真。如果  $N_A > N_B$ ,那么大约为 1/2 的概率,我们有

$$\sigma = \text{Padding}(M, r)^{d_A} \pmod{N_A} > N_B$$

但是,由于  $|N_A| = |N_B| = k$ ,所以有

$$\sigma < N_A < 2^k$$

因此,令

$$\sigma' \leftarrow \sigma - 2^{k-1}$$

即  $\sigma'$  是“砍掉” $\sigma$  最高位比特的结果,则

$$\sigma' < 2^{k-1} < N_B$$

也就是说,Bob 可以正确恢复  $\sigma'$ 。在此之后,Bob 的验证步骤会指导 Bob 是否需要将“砍掉”的比特修正回来。

### 16.5.2.1 RSA-TBOS

Malone-Lee 和 Mao[184]的 RSA-TBOS 方案应用了 PSS-R 填充方案(见 16.4.4 节),该签密方案在算法 16.4 中具体描述。

签密中的第 6 步是为了确保  $c' = N_B$ 。如果最初  $c'$  不能通过该测试,那么我们有  $N_A > c' > N_B$ 。因为  $N_A$  和  $N_B$  都有  $k$  比特,我们可以推出  $c'$  也是  $k$  比特,所以赋值  $c' \leftarrow c' - 2^{k-1}$  等于将  $c'$  的最高位比特去掉。这样就有我们要求的  $c' < N_B$ 。

注意这个步骤可能会导致解签密中的额外步骤。特别地,可能需要执行  $c'^{e_A} \pmod{N_A}$  两次(两个  $c'$  相差  $2^{k-1}$ )。定义另一种在签密阶段进行试错测试的方案是可能的。这意味着对不同的  $r$  值重复签密步骤 1~5 直到得到一个  $c' < N_B$ 。

不可否认性对 RSA-TBOS 来说是非常简单的。签密的接收者执行解签密过程,直到步骤 2,此时可以将  $c'$  交给一个可以验证其有效性的第三方。

虽然 TBOS 签密方案有很多吸引人的特性(在描述该算法之前就已经列举过了),但我们应该注意因为应用了 RSA-PSS-R 填充方案,它继承了一个缺点:恢复消息的带宽太低。读者可以回顾在 RSA-PSS-R 填充方案中我们关于这一点的讨论(见 16.4.4.2 节)。

**算法 16.4 一箭双雕:RSA-TBOS 签密方案****密钥参数**

假设  $k$  是一个偶正整数。发送者 Alice(接收者 Bob)的 RSA 公私钥材料分别是  $(N_A, e_A)$ ,  $(N_A, d_A)[(N_B, e_B), (N_B, d_B)]$ , 满足  $|N_A| = |N_B| = k$ 。

假设  $G$  和  $H$  是两个杂凑函数, 满足

$$H: \{0,1\}^{n+k_0} \mapsto \{0,1\}^{k_1} \text{ 和 } G: \{0,1\}^{k_1} \mapsto \{0,1\}^{n+k_0}$$

其中  $k = n + k_0 + k_1$ ,  $2^{-k_0}$  和  $2^{-k_1}$  都是可忽略的量。

**签密**

当 Alice 为 Bob 签密消息

$M \in \{0,1\}^n$  时, 她执行:

1.  $r \leftarrow_U \{0,1\}^{k_0}$
2.  $\omega \leftarrow H(M \parallel r)$
3.  $s \leftarrow G(\omega) \oplus (M \parallel r)$
4. If  $s \parallel \omega > N_A$  goto 1
5.  $c' \leftarrow (s \parallel \omega)^{d_A} \pmod{N_A}$
6. If  $c' > N_B$ ,  $c' \leftarrow c' - 2^{k-1}$
7.  $c \leftarrow c'^{e_B} \pmod{N_B}$
8. Send  $c$  to Bob

**解签密**

当 Bob 要对来自 Alice 的

密文  $c$  解签密时, 他执行:

1.  $c' \leftarrow c^{d_B} \pmod{N_B}$
2. If  $c' > N_A$ , reject
3.  $\mu \leftarrow c'^{e_A} \pmod{N_A}$
4. Parse  $\mu$  as  $s \parallel \omega$
5.  $M \parallel r \leftarrow G(\omega) \oplus s$
6. If  $H(M \parallel r) = \omega$ , return  $M$
7.  $c' \leftarrow c' + 2^{k-1}$
8. If  $c' > N_A$ , reject
9.  $\mu \leftarrow c'^{e_A} \pmod{N_A}$
10. Parse  $\mu$  as  $s \parallel \omega$
11.  $M \parallel r \leftarrow G(\omega) \oplus s$
12. If  $\omega \neq H(M \parallel r)$ , reject
13. Return  $M$

**16.5.2.2 安全性证明**

Malone-Lee 和 Mao 对 TBOS 签密方案给出了几个强安全特性的形式化归约证明[184]。他们还给出了安全签密模型的描述。这些强安全特性是: 在 IND-CCA2 模型下的保密性和在选择消息攻击下的签名不可伪造性。

因为这些证明与我们在 15.2 节中给出的 RSA-OAEP 证明类似, 而且详细描述也很繁琐, 所以在这里不再给出这些归约。对此有兴趣的读者可以参阅[184]中的详细情况。

但是, 即使没有描述这些归约的细节, 我们仍然可以对以下问题有一个非形式化的、抽象的理解: 为什么使用 RSA 函数陷门性的、可证明安全的加密填充方案能够构成一个安全的签名方案。显然, 我们需要证明选择消息攻击下的签名不可伪造性。现在我们就尝试使用(大家早已熟知的)OAEP 填充方案理解这一点。

回忆抵抗 IND-CCA2 模式下攻击的 RSA-OAEP 归约证明(在 15.2 节中给出)。那里我们估计过如果 Malice 不遵从预先描述的加密过程, 那么不管 Malice 使用的是什么算法(回忆, Malice



是一个黑盒),也不管他可以适应性地构造密文(即在适应性选择密文解密训练背景下)这个事实,他可以提交一条有效密文的概率在统计上都是可忽略的。

上述事实可以机械地转换为对随机化填充签名方案签名不可伪造性的证明:即使是在适应性选择消息训练背景下,如果不使用预先描述过的签名过程(因为没有签名指数),Malice 伪造一个有效消息-签名对(与以下伪造对应:不使用预先描述的加密过程,构造有效的明文-密文对)的概率在统计上是可忽略的。

当然,虽然在直觉上令人信服,但是我们必须强调,这个描述并不是对基于 RSA 填充签名方案安全性的形式化证明,因为它没有遵从我们建立的“归约为矛盾”的形式化方法。有兴趣的读者可以在[184]中查阅这个归约证明。

## 16.6 本章小结

本章开始我们介绍了数字签名方案的强安全性定义:在适应性选择消息攻击下的签名不可伪造性。签名方案的这种攻击模式与公钥加密方案的 IND-CCA2 模式对应。这两种模式共同的基本思想是在这些攻击中允许 Malice 进行密码分析训练。说一个密码体制是强的,如果它可以抵抗攻击,即使在攻击中为 Malice 提供密码分析训练,而且即使他希望做多少训练就做多少训练[只要他是多项式有界的,而且(因此)训练会话中交互次数也是多项式有界的]。

接着我们研究了两个重要的适于应用的签名方案族。第一个是三元组 ElGamal 族签名方案,第二个是应用于陷门单向置换中的随机化填充方案,其中的陷门单向置换可以是 RSA 和 Rabin 函数等。

然后我们进一步对这两族中的签名方案建立了强安全性的形式化证明。

对于第一族方案,我们学习了基于 ROM 的归约证明技术,该技术在下述原理下工作:成功“分叉伪造者的询问”的概率是不可忽略的,也就是说,对于伪造者的一组询问可以有两组完全不同的答案,而且两组回答都服从正确的随机分布(均匀分布),从这个意义上说这两组回答都是正确的。所做的询问被分支的伪造者是一个无意识的概率算法,所以只要满足正确分布它就工作。这样,虽然对询问的回答分了叉,但这个被分叉的伪造者并没有被愚弄,而且它会因此而继续帮助归约算法解决一个难问题:离散对数问题。我们还描述了对于这一族签名方案的另一种证明方法:重行模型。虽然正如我们所分析的,这两种证明方法都是严格形式化的,但是它们都不是非常有效。因此,证明只对相当大的安全参数有意义。

对于第二族方案,其中的签名方案都是连续组合用于陷门单向置换的随机化填充而得到的。对它们基于 ROM 的归约证明,与由单向置换中的随机化填充构造的公钥加密方案的证明类似,我们在前一章已经学习过对后者的证明。但是,对于签名的情况,成功的攻击(在适应性选择消息攻击下伪造签名)会导致对单向函数的直接求逆。因此,对基于随机化填充签名方案得到的归约证明是严谨的,即攻击者伪造签名的能力可以完全地转换为对一个困难问题求逆(即方案基于的陷门单向函数)的能力。这就叫做具体的安全特性。

最后,我们还学习了有效而且有用的密码学原型——签密方案。与本书介绍的其他适于应用的加密和签名方案类似,本章介绍的签密方案也是基于两个熟知的密码学基本问题:离散对数问题和整数分解问题。



## 习题

- 16.1 对于数字签名,“适于应用”的安全性定义是什么?
- 16.2 假设 Malice 是一个坏人,那么我们为什么给他权利,让他得到他选择消息的签名,而且他还希望能得到多少就得到多少?
- 16.3 在对三元组 ElGamal 签名安全性基于 ROM 的分叉引理证明中,Simon 运行 Malice 两次并对同一组 RO 提问给出两组相互独立的回答。我们可以认为在第二次运行中 Simon 欺骗了 Malice 吗?
- 16.4 讨论三元组 ElGamal 签名的存在性伪造对该方案的安全性证明所起到的作用。
- 16.5 使用 PSS 对同一消息签名两次,该算法输出同一签名值的概率有多大?
- 16.6 在习题 15.2 中,我们定义了加密方案的带宽。对消息恢复数字签名方案的带宽,定义类似。在与习题 15.2 中的安全参数设置相同的情况下,使用通用的 RSA 填充方案(算法 16.2)对 (i) 签名、(ii) 加密的带宽是多少?
- 16.7 上述问题中的两个带宽结果为什么不相同?
- 16.8 讨论 Zheng 签密方案提供的不可否认性和 TBOS 签密方案提供的不可否认性有什么不同。
- 16.9 我们对 TBOS 签密方案不可伪造性的论证(见 16.5.2.2 节)是令人信服的,但它是不是一个形式化的安全性证明? 为什么?  
提示:该论证是一个归约证明吗?

## 第 17 章 分析认证协议的形式化方法

### 17.1 引言

在第 11 章我们目睹了这样一个事实:认证和认证的密钥建立协议(本章中的认证协议通常是指这两种协议)都以易于出错著称。这些协议会以令人非常吃惊的方式出错。怎样设计认证协议才能使它们安全,是具有各种背景的研究者们正在从事的一个严峻课题。这些研究者包括密码学家、数学家以及理论计算机科学家。他们一致认为在认证协议的分析中应该采用形式化方法。

形式化方法是对非形式化方法的自然扩展。形式化内涵丰富,包括各种概念,譬如系统的方法、机械的方法、由规则和/或工具支持的方法。形式化方法通常支持用符号表示的一个系统或一种描述语言,这种描述语言模型化并详细说明系统的行为,这就使我们可以获得(即可以理解)系统的行为,而且可以通过严格的应用逻辑和数学方法推理出来。形式化方法有时是一个专家系统,它理解人类的经验甚至试图模型化人类的智慧。形式化方法的一个普遍特征是在解决问题时采用系统的方法,有时使用穷举方法。因此,形式化方法对于分析复杂系统尤其适用。

在形式化分析认证协议的领域内,我们可以分为两种不同的方法。一种是对某个渴望性质或者安全特性形式化推理;另一种是对某些不良或者不安全特性的系统查找。

在第一种方法中,要分析的协议必须精心挑选或设计,它应该是人们已经相信是正确的或者很可能是正确的协议。我们的分析力图证实对于已被仔细形式化、希望具有的一组性质来说,该协议确实是正确的。因为要分析的协议是精心挑选的,所以虽然形式化证明方法可以更一般,但形式化证明通常是特定目标协议量身定做的,因此可能需要涉及很多人类智慧。这一方法又分为两个学派:计算学派和符号操作学派。在计算学派中,安全特性是以概率的形式定义和度量的,而安全性或协议正确性的证明是数学证明某个定理成立,通常涉及到一个归约变换,归约到一个广泛接受的复杂性理论假设(关于可证明安全公钥加密方案的证明,见第 14 章和第 15 章)。在符号操作学派中,这一学派的成员是形式化方法领域内的理论计算机科学家,安全特性被表示成一组抽象的可操作符号,符号操作有时由一个形式化逻辑系统进行,有时由一个称为定理证明器的机械工具进行,该证明器最后要回答 YES/NO。

第二种方法认为,一个认证协议即使是精心挑选或设计的,甚至经过了一个正确性的形式化证明(即作为第一种方法的结果),仍然可能含有错误。这是因为“正确性的证明”只能表明一个协议满足一组具体指定的渴望性质,如果在“安全性证明”过程中没有考虑到某一种失败可能,这个可证明安全的协议还是可能失败的。因此,在这种方法中,分析的方式就是系统或穷举搜索错误。协议的形式化就是将协议表示成一个(有限的)状态系统,状态系统通常由不同参与者各自运行的那一部分协议(包括“Malice 那一部分”)的子状态系统组成。错误可以用通常的方式描述,例如,在消息秘密性的情况下,不好的状态可以是该消息最后在 Malice 的知识集内;在实体认证的情况下,不好的状态可以是一个错误身份最后在接收诚实主体的身份

集内。这种方法与理论计算机科学中复杂系统的形式化分析领域有着密切的关系,因此它会经常使用该领域中比较完善的自动分析工具。

本章我们将学习这些方法,对认证协议进行形式化分析。

### 17.1.1 本章概述

本章的技术部分以形式化协议描述开始;在 17.2 节,我们将学习描述认证协议的精确方法。在协议描述这个主题之后,我们将集中讨论分析技术。17.3 节介绍一种证明技术,它基于协议正确性的计算模型,在该计算模型中,证明的形式是用数学方法证明一个定理成立。在 17.4 节,我们将介绍一种通过操作抽象符号证明协议安全性的技术、一种逻辑方法和一种定理证明方法。17.5 节介绍形式化分析技术,它把协议模型化为一个有限状态系统,然后查找系统错误。最后,在 17.6 节,我们将简要讨论新近的一种研究,它试图将计算观点下的安全性和符号观点下的安全性联系起来。

## 17.2 认证协议的形式化描述

让我们开始本章的技术部分:给出认证协议需要更形式化的描述方法的证据。在分析复杂系统的任何形式化方法中,描述都应该是一个不可缺少的组成部分。当复杂系统是认证协议时,我们认为学习的范围应该包括用一种更精确的描述方式来描述密码学变换的使用。

正如我们在第 2 章和第 11 章中看到的,很多认证协议的设计中只使用了加密,所以要表示这些协议中加密的使用,一种广泛认可的符号是  $\{M\}_K$ 。这个符号表示一条密文,它的发送者必须执行加密来生成它,而它的接收者为了从它提取  $M$  就必须执行解密。正是这些密码学能力的展示,才向通信伙伴证明了一个主体持有密钥,从而证明该主体的身份。

因此,使用加密获得认证的想法似乎很简单,没什么微妙之处。

但是,事实上,使用密码学变换获得认证的这个简单想法经常被误用。这种误用是很多协议存在缺陷的原因。本节我们首先确认认证协议中对加密的一种普遍误用;然后基于对使用密码学变换的一种精确描述,我们提出一种设计认证协议的方法。

### 17.2.1 加解密认证方法的不精确性

在 11.4.1.5 节中我们列举了两种使用加密构造认证协议的“非标准”方法。在这两种方法中,发送者生成密文  $\{M\}_K$  并发送给意定的接收者;正确的接收者有私钥可以执行解密,从而可以把从这条密文提取的消息成分返回给该发送者。返回的这个消息成分通常包含新鲜性标识符,向发送者证明了同接收者真实的通信关系。这就达到了发送者对接收者的认证。

我们称这些(非标准)机制为“加解密认证”方法。

在加解密方法中,我们通常不声明但却暗中起作用的安全服务是保密性,这必须用一个可逆密码学变换来实现。但是,在很多使用该方法的认证协议中,需要的安全性服务其实不是保密性,而是数据完整性,某些单向(即不可逆的)密码学变换就可以很好地实现这一点。这就是为什么我们把这种情况归类为误用密码学变换的原因。

当发生密码学变换误用时,会出现两个不希望的后果。现在我们就对它们进行详细讨论。

### 17.2.1.1 危害

在验证消息新鲜性的询问-应答机制中,加解密方法有利于敌手利用参与者来得到解密预言服务(见 7.8.2.1 节和 8.9 节)。这种服务可能会授予 Malice 一个未授权的密码操作;否则,由于 Malice 没有正确的密钥,他无权进行这个密码操作。

预言解密服务为 Malice 操作协议的消息并挫败认证的目的提供了主要的技巧来源。对 Needham-Schroeder 公钥认证协议的 Lowe 攻击(攻击 2.3)就用了这样的技巧:在攻击步骤 1~7 中,Malice 使用 Alice 的预言解密服务解密 Bob 的一次性随机数,然后他就可以假冒 Alice 与 Bob 对话。

预言解密服务还为 Malice 提供了可用于密码分析的有价值信息,譬如在选择明文或选择密文攻击中。我们在第 14 章的许多攻击实例中都看到了这种技巧。

在基于询问-应答的机制中,接收者为了证明密码学凭证(拥有正确密钥),他/她应该使用的正确密码学变换是单向变换。在使用对称密码技术的情况下,机制(11.4.2)更受欢迎。如果新鲜性标识符必须保密,那么可以使用机制 11.4.1,但在这种情况下,Bob 仍然需要用数据完整性服务来保护他的密文(原因将在 17.2.1.2 节中给出),事实上这需要使用单向变换来得到,即机制 11.4.1 中的密文仍然需要基于机制 11.4.2 的保护。在使用非对称加密技术的情况下,标准的机制是机制 11.4.3。

当然,机制 11.4.1 和机制 11.4.2 也可以使询问者利用应答者提供的预言服务,生成明密文对:

$$\begin{aligned} N, \mathcal{E}_K(\dots, N) \\ N, \text{MDC}(K \dots, N) \end{aligned}$$

其中  $N$  是询问者选择的新鲜性标识符。但是,由于  $N$  是非秘密参数,提供这样的明密文对所引起的问题远比提供解密服务引起的要少。

而且,在第二种情况下(机制 11.4.2),事实上并不能得到“预言解密服务”。任何实现 MDC 的单向密码学变换都有数据压缩的特性(例如回顾 10.3.1 节和 10.3.3 节中关于基于杂凑函数和基于分组密码的 MDC 的数据压缩特性)。数据压缩特性会导致信息损失,这正是这种变换不可逆的原因。信息损失使得到的询问/应答对在其他不同的情况下是不可用的:它们的使用只限于机制 11.4.2 的情况;其他情况下的使用都会引起可以检测到的错误。

### 17.2.1.2 不充分性

通常,作为保密信息的加密,密文自身也要利用消息完整性保护。如果该密文是一条重要的协议消息,那么缺少消息完整性保护就不可能阻止主动攻击者篡改加密后的信息并挫败协议的目标。

现在,以 Needham-Schroeder 对称密钥认证协议(Denning 和 Sacco 修正后的版本,见 2.6.5.1 节)为例,我们来看看这个问题。假设该协议中使用的加密算法为密文包含的所有消息成分都提供了强的秘密性保护。但是,为了阐明我们的观点,我们将约定加密算法没有提供任何消息完整性保护。这个约定并不是没有道理的。事实上,如果明文消息含有足够数量的随机性使得解密提取的明文不可识别,那么只要加密算法没有设计成还能提供消息完整性保护,它就可以具有该特性。

例如,我们可以合理地假设该加密算法是 CBC 操作模式(见 7.8.2 节)下的 AES(见 7.7 节)。读者可以将我们的攻击扩展到其他对称加密算法,如一次一密。应该注意,不管使用什么样的加密算法,我们的攻击都不会利用算法提供保密性服务时任何质量上的缺陷。

现在我们检查 Needham-Schroeder 对称密钥认证协议的开始两步。

1. Alice  $\rightarrow$  Trent:  $Alice, Bob, N_A$ ;
2. Trent  $\rightarrow$  Alice:  $\{N_A, K, Bob, Y\}_{K_{AT}}$ ;

其中  $Y = \{Alice, K, T\}_{K_{BT}}$ 。

令明文消息串

$$P_1, P_2, \dots, P_\ell$$

对应的明文消息分组是

$$N_A, K, Bob, Y$$

为了使该协议适合一般应用的需要,我们合理地假设会话密钥  $K$  比一个密文分组长或相等。这确实是合理的,因为会话密钥含有足够多的信息比特(例如为了分组密码密钥和流密码种子密钥的安全)。为了防止预测,一次性随机数  $N_A$  也必须足够大。因为  $N_A$  从  $P_1$  开始,所以我们对会话密钥大小的假设自然会得出整个明文分组  $P_2$  完全包含会话密钥,或者可能只包含会话密钥的一部分。

注意,虽然我们将  $P_2$  和  $K$  联系起来,但这纯粹是为了清楚表述,如果会话密钥  $K$  非常大,那么它就可能占用从  $P_2$  开始的多个明文分组。当然,Malice 会知道会话密钥  $K$  的大小。事实上,我们的攻击要求 Malice 知道明文消息的大小和实现的细节。毕竟,这些都不是应该保密的。

令

$$IV, C_1, C_2, \dots, C_\ell$$

表示明文分组  $P_1, P_2, \dots, P_\ell$  对应的 AES-CBC 密文分组(复习 7.8.2 节中的 CBC 操作模式)。再令

$$IV', C'_1, C'_2, \dots, C'_\ell$$

表示同一对参与者之间的前一次协议(同一个协议)运行得到的密文分组。当然,Malice 记录了这个旧密文分组。

为了在新的一次运行中攻击该协议,Malice 要截取 Trent 发给 Alice 的新密文分组:

2. Trent  $\rightarrow$  Malice("Alice"):  $IV, C_1, C_2, C_3, \dots, C_\ell$ ;

Malice 现在以下列方式替换这些分组:

2. Malice("Trent") Alice:  $IV, C_1, C'_2, C'_3, \dots, C'_\ell$

也就是说, Malice 将目前运行的最后  $\ell - 1$  个分组分别替换为他(从这个协议的以前某次运行中)记录过的旧分组,并将这串篡改过的分组链发送给 Alice,使其看上去似乎是从 Trent 发出的。

Alice 的 CBC 解密会以正确的顺序返回  $N_A$ , 因为解密结果是  $IV$  和  $C_1$  的函数。解密将返回(回顾 7.8.2 节中的“CBC 解密”)



$$\hat{K} = D_{K_{AT}}(C'_2) \oplus C_1 = K' \oplus C'_1 \oplus C_1$$

作为“新”的会话密钥(或“新”会话密钥的第一个分组)。这里  $K'$  是该协议以前某次运行记录中的旧会话密钥(或旧会话密钥的第一个分组)。Alice 对接下来的密文分组的解密会返回其余的  $\ell - 1$  个明文分组,与她过去运行该协议得到的相同。

因为  $K'$  是旧的会话密钥,我们不能排除 Malice 已经以某种方式得到它的可能性(可能是因为 Alice 和 Bob 偶然泄露了)。所以, Malice 可以使用  $\hat{K}$  (如果会话密钥的长度比单个组长,它可能是  $\hat{K}$  与除  $K'$  以外的其余分组级联的结果)假冒 Bob,与 Alice 对话。

从这个攻击我们可以看到,不管 Alice 从她正确提取的新鲜性标识符  $N_A$  推断出什么,我们都不能认为 Alice 解密操作返回的其他明文消息是新鲜的!

有很多方法可以实现这个协议中的加解密,每一种都可能可以阻止这种特殊攻击。但是,只要对 Malice 不保密实现细节,它们就可能遭受其他的不同攻击。

在两个早期的国际标准草案中,有几个认证协议[146,147]沿用了错误的思想——用 CBC 实现加密以提供数据完整性服务(对于这些使用 CBC 的协议,一般的指南可以参考文献[148,144]),当然,就像我们在本节中介绍的,这些协议都有致命的缺陷[186,187]。

我们相信使 Needham-Schroeder 对称密钥认证协议安全的正确方法是用正确的数据完整性服务保护密文分组,例如,应用我们在 10.3.2 节和 10.3.3 节中介绍的消息认证技术(篡改检测码技术)。这种技术本质上基于单向变换,而不是加密解密方法。

到现在为止,我们清楚地表明了在使用对称密码学技术的认证协议中,为了使认证协议安全,加密解密方法是不充分的。

在使用非对称密码技术的认证协议中,加密解密方法也是不充分的。Needham-Schroeder 公钥密码认证协议(协议 2.5)是这种方法的一个例子。Lowe 对该协议的攻击(攻击 2.3)给出了这种不充分性的明显证据。我们在后面(17.2.3.3 节)会看到,对于防止 Lowe 攻击来说,描述这个协议的单向变换方法给出了对该协议的一个良好修正。

## 17.2.2 认证协议的细化描述

为了描述认证协议,以便精确地表达需要的密码学服务,Boyd 和 Mao 提出用一种更完备的方式描述认证协议[188]。他们采用一种细化的方法,利用两个符号表述了密码学变换的用途。这两个符号是:

- $\{M\}_K$  表示对消息  $M$  使用密钥  $K$  的加密。它对  $M$  提供的安全性服务是保密性:只有拥有与  $K$  匹配的解密密钥  $K^{-1}$  的主体可以提取  $M$ 。注意解密过程输出的消息对于  $K^{-1}$  的持有者可能是不可识别的。
- $[M]_K$  表示使用密钥  $K$  对消息  $M$  单向变换的结果。它对  $M$  提供的安全性服务是具有消息源识别的数据完整性,需要用我们在第 10 章中学习过的技术。 $[M]_K$  中的消息  $M$  不是秘密,而且可能即使不执行任何密码学操作也能从  $[M]_K$  看出  $M$ 。拥有与  $K$  匹配的验证密钥的主体  $K^{-1}$  可以验证  $[M]_K$  的数据完整性并识别消息源。验证过程输出 YES 或 NO:在 YES 的情况下,我们认为  $[M]_K$  具有正确的数据完整性,因此将  $M$  看做是由被识别的源发送的可识别消息;在 NO 的情况下,我们认为  $[M]_K$  具有不正确的数据完整性,因此将  $M$  看做是不可识别的消息。



在实际中,  $[M]_K$  可以通过一个  $(M, \text{prf}_K(M))$  对实现, 其中  $\text{prf}_K$  在对称加密技术的实现中表示一个带密钥的伪随机函数(例如在密码分组链接操作模式下的消息认证码, CBC-MAC, 见 10.3.3 节, 或带密钥的密码学杂凑函数, HMAC, 见 10.3.2 节), 在非对称加密技术的实现中表示一个数字签名算法, 这些都是高效实用的实现方式。

这两个精确的符号统一了对称和非对称密码学技术。在前一种情况下,  $K$  和  $K^{-1}$  是相同的, 而在后一种情况下, 它们是公钥密码算法中匹配的密钥对。

我们要强调变换  $[M]_K$  不仅提供数据完整性服务, 而且提供消息源识别服务。如果对  $[M]_K$  的验证返回 YES, 那么即使消息  $M$  可能不含有它的源的任何信息, 验证者也能基于正在使用的验证密钥识别正确的源。

最近, 人们越来越广泛地接受密文(保密性服务)和数据完整性相结合的服务的重要性。我们在第 15 章中看到了这个想法在公钥密码学中的一般应用。在安全性领域内, Aiello 等[11]使用下面的符号精炼描述安全服务:

$$\{M\}_{K_1}^{K_2}$$

在这个符号中, 对消息  $M$  用  $K_1$  加密, 同时  $M$  受到数据完整性保护, 其中  $K_2$  是验证密钥。

### 17.2.3 认证协议细化描述的例子

现在, 我们介绍几个用细化符号描述的认证协议。

#### 17.2.3.1 Needham-Schroeder 对称密钥认证协议

第一个例子是 Needham-Schroeder 对称密钥认证协议, 它的具体描述在协议 17.1 中。

---

#### 协议 17.1 Needham-Schroeder 对称密钥认证协议的细化描述

假定与目标: 与协议 2.4 相同。

1. Alice  $\rightarrow$  Trent: Alice, Bob,  $N_A$ ;
  2. Trent  $\rightarrow$  Alice:  $[\{K\}_{K_{AT}}, N^A, \text{Alice}, \text{Bob}]_{K_{AT}};$   
 $[\{K\}_{K_{BT}}, T, \text{Alice}, \text{Bob}]_{K_{BT}};$
  3. Alice  $\rightarrow$  Bob:  $[\{K\}_{K_{BT}}, T, \text{Alice}, \text{Bob}]_{K_{BT}};$
  4. Bob  $\rightarrow$  Alice:  $[N_B]_K;$
  5. Alice  $\rightarrow$  Bob:  $[N_B - 1]_K。$
- 

在 Needham-Schroeder 对称密钥认证协议的细化描述中, 需要数据完整性服务这一点非常清楚。如果 Alice 和 Bob 从 Trent 那里接收到的消息以及他们之间发送的消息在传输过程中没有被更改, 那么在双方看到数据完整性检验输出的 YES 之后, 都可以相信会话密钥  $K$  和他们的身份以及他们各自的新鲜性标识符之间有正确的密码学联系。这使得他们确信共享密钥双方的正确性以及会话密钥的新鲜性。很明显, 对消息任何未授权的更改, 如我们在 17.2.1.2 节中看到的, 都会检测出来。

从该协议的这个细化描述,我们可以看到它以最起码的标准提供了保密性服务:其中只有会话密钥  $K$  受到保护。因为密钥本质上是随机的,所以最小程度地使用保密性服务是我们渴望的,这样可以限制(最小化)可能有利于密码分析的信息泄漏的数量。

### 17.2.3.2 Woo-Lam 协议

第二个细化协议描述的例子是对 Woo-Lam 协议的描述。这在协议 17.2(参考协议 11.2)中给出。该例描述将揭示使用细化描述对避免各种攻击非常有效。我们已经看到的对于该协议的那些攻击,之所以奏效的原因将变得非常明显:在认证协议中,人们广泛接受的用以表达加密用途的不精确符号隐含了不正确的密码学服务。

我们注意到 Woo-Lam 协议(协议 11.2)中的消息都是不保密的,所以没有必要对任何协议消息提供机密性保护。该协议中需要的密码学保护只是具有源识别的数据完整性。因此,在该协议的细化描述中,我们只描述单向变换。

对最初的 Woo-Lam 协议以及我们在 11.7 节中看到的各种改进协议的攻击都不能应用于该协议的细化描述形式,现在我们就给出这个的原因。

#### 协议 17.2 Woo-Lam 协议的细化描述

前提与目标:与协议 11.2 相同。

约定:

任何一个单向变换验证返回 NO 就中止运行协议。

1. Alice  $\rightarrow$  Bob:  $Alice$ ;
2. Bob  $\rightarrow$  Alice:  $N_B$ ;
3. Alice  $\rightarrow$  Bob:  $[N_B]_{K_{AT}}$ ;
4. Bob  $\rightarrow$  Trent:  $[Alice, N_B, [N_B]_{K_{AT}}]_{K_{BT}}$ ;  
( \* 注意, Bob 的单向变换部分包含  $N_B$ , 因为他自己就是这个新鲜性标识符的源 \* )
5. Trent  $\rightarrow$  Bob:  $[N_B]_{K_{BT}}$ ;
6. 如果  $[N_B]_{K_{BT}}$  的完整性检验返回 YES, Bob 就接受, 否则拒绝。

首先,在攻击 11.5 中给出的并行会话攻击不会起作用。为了证明这一点,我们假设 Malice 在两个并行步骤 3 和 3' 中将  $[N_B]_{K_{MT}}$  发送给 Bob:

3. Malice("Alice")  $\rightarrow$  Bob:  $[N_B]_{K_{MT}}$
- 3'. Malice  $\rightarrow$  Bob:  $[N_B]_{K_{MT}}$

我们假设 Bob 运气仍然不好:他没有检查这些消息(因为协议中并不要求他检查这些消息),而只是继续在两个并行的步骤 4 和 4' 中发送两个消息:

4. Bob  $\rightarrow$  Trent:  $[Alice, N_B, [N_B]_{K_{MT}}]_{K_{BT}}$

4'. Bob  $\rightarrow$  Trent:  $[Malice, N'_B, [N_B]_{K_{MT}}]_{K_{BT}}$

但是, Trent 会在这两个步骤中检测到这两个错误。第一个错误发生在对第 4 步中的消息进行验证时: Trent 使用  $K_{AT}$  验证  $[N_B]_{K_{MT}}$ , 这当然返回 NO, 所以第 4 步的运行会停止。第二个错误发生在对第 4' 步中的消息进行验证时: Trent 发现两个单向变换使用的一次性随机数不同, 因此运行也必然停止(不然的话, Trent 把两个一次性随机数中的哪一个返回给 Bob 呢?)

最后, 不难看出攻击 11.6 中的反射攻击也不会对细化描述起作用。这是因为如果 Malice 在消息行 5 中反射消息 4, 那么 Bob 在步骤 6 中执行的验证当然会返回 NO。

现在很清楚了, 最初的 Woo-Lam 协议存在缺陷的根本原因就是它对密码学服务的误用。

### 17.2.3.3 Needham-Schroeder 公钥认证协议

最后让我们看看有关公钥应用的例子: 对 Needham-Schroeder 公钥认证协议的细化描述。

由 2.6.6.3 节中的讨论, Needham-Schroeder 公钥认证协议可以表示成三步消息传输。简化形式在协议 17.3 中给出。

#### 协议 17.3 Needham-Schroeder 公钥认证协议

前提与目标: 与协议 2.5 相同。

1. Alice  $\rightarrow$  Bob:  $\{N_A, Alice\}_{K_B}$ ;
2. Bob  $\rightarrow$  Alice:  $\{N_A, N_B\}_{K_A}$ ;
3. Alice  $\rightarrow$  Bob:  $\{N_B\}_{K_B}$ 。

这里  $\{\dots\}_{K_A}$  和  $\{\dots\}_{K_B}$  分别表示使用 Alice 和 Bob 的公钥加密, 所以它们只能分别由 Alice 和 Bob 解密。这样, 一旦收到消息行 2 中的消息, Alice 就应该相信只有 Bob 可以得到消息行 1 中消息的解密, 然后返回她的一次性随机数  $N_A$ 。同样地, 一旦收到消息行 3 中的消息, Bob 就应该相信只有 Alice 可以得到消息行 2 中消息的解密, 然后返回他的一次性随机数  $N_B$ 。所以, 我们非常合理地期望运行一结束双方就都已经真实地识别对方并共享了两个一次性随机数。

但是, 攻击 2.3 中的 Lowe 攻击挫败了这个“合理的”期望。17.2.1 节中我们令人信服地证明了通过加密解密方法获得的认证不精确之后, 就可以从另一个新的角度回顾对最初协议的 Lowe 攻击: 造成攻击成功的原因正是缺少正确的密码服务。如果用细化的精确方式描述该协议, 那么这个攻击就会消失。协议 17.4 就是一个例子。

#### 协议 17.4 Needham-Schroeder 公钥认证协议的细化描述

前提与目标: 与协议 2.5 相同。

1. Alice  $\rightarrow$  Bob:  $\{[N_A, Alice]_{K_A}\}_{K_B}$ ;
2. Bob  $\rightarrow$  Alice:  $\{N_A, [N_B]_{K_B}\}_{K_A}$ ;
3. Alice  $\rightarrow$  Bob:  $\{[N_B]_{K_A}\}_{K_B}$ 。

在这个细化的描述中,  $[N_A, Alice]_{K_A}$  表示一个消息, 它的数据完整性检验(和消息源识别)必须使用 Alice 的公钥  $K_A$ 。因此, 变换  $[N_A, Alice]_{K_A}$  可以是 Alice 的签名。所以, 第 1 步中的消息是 Alice 的一次性随机数, 由 Alice 签名后再用 Bob 的公钥加密。同样, 第 2 步中的消息可以是 Bob 先签名再用 Alice 的公钥加密的消息。

因为第 2 个消息由 Bob 签名, 所以攻击 2.3 中的 Lowe 攻击对细化形式的协议不再起作用。Malice 可以通过使用 Alice 的签名(Alice 将她的签名发给了 Malice, Malice 可以解密恢复该签名并使用 Bob 的公钥再加密)假装 Alice 发起与 Bob 的一次运行, 但是, 现在 Malice 不能像对最初协议的攻击中所做的那样, 将 Bob 的应答再发给 Alice, 因为当 Alice 验证 Malice 的签名时, 她会检测到错误。

### 协议 17.5 Needham-Schroeder 公钥认证协议的另一种细化描述

前提与目标: 与协议 2.5 相同。

1. Alice  $\rightarrow$  Bob:  $[\{N_A\}_{K_B}, Alice]_{K_A}$ ;
2. Bob  $\rightarrow$  Alice:  $[\{N_A, N_B\}_{K_A}]_{K_B}$ ;
3. Alice  $\rightarrow$  Bob:  $[\{N_A\}_{K_B}]_{K_A}$ 。

针对 Needham-Schroeder 公钥认证协议的 Lowe 攻击, 虽然在 2.6.6.4 节中建议通过在加密的消息中增加预期通信伙伴的身份来改进, 但是现在我们知道这一修补未必正确。事实上, 如果加密算法可展(见 14.5.3 节), 解密者就不能确信解密所揭示身份的正确性。显然, 正确的改进要用正确的密码学服务, 而细化的协议描述有助于我们识别和描述正确的服务。

可以用另一种方法细化 Needham-Schroeder 公钥认证协议, 先加密, 后签名, 如协议 17.5 所描述。

最初协议的 Lowe 攻击对协议 17.5 也不起作用: 现在 Malice 甚至不能假冒别人发起与 Bob 的一次协议运行。

## 17.3 正确协议的计算观点——Bellare-Rogaway 模型

在第 14 章和第 15 章我们熟悉了计算模型下可证明安全性的思想, 它源自 Goldwasser 和 Micali 的开创性研究[127]。那里, 安全特性(几个保密性质之一)是在给定攻击背景(几个攻击游戏之一, 其中每一个都以充分的一般性和精确性模型化了现实世界中公钥加密方案攻击者的几种典型行为中的某一种行为)下讨论的。针对一个所谓的攻击, 对公钥加密方案的安全性证明就是给出一个有效的变换(称为多项式时间的归约), 将所谓的攻击转变成为一个计算复杂性理论中人们广泛相信困难的问题的重大突破。正是由于人们广泛相信这个重大突破不可能, 才导致了与所谓攻击的存在性相矛盾, 也就是说, 证明是通过矛盾给出的。

因此, 在计算模型下安全性的形式化证明包括以下三步:

- i) 形式模型化协议参与者和攻击者的行为: 该模型化通常是以攻击者和攻击目标之间进行攻击游戏的形式给出的。

- ii) 安全性目标的形式化定义: 这里定义攻击者在攻击游戏中的成功, 通常以(不可忽略的)概率和(可承受的)时间复杂度公式的形式给出。
- iii) 形式化证明一个多项式时间的归约, 把对给定目标的所谓攻击归约到计算复杂性理论中的一个认为不可能的突破; 该归约的形式化证明是数学上证明一个定理成立。

Bellare 和 Rogaway 是最初的研究者, 他们最先使用计算模型方法证明认证和认证的密钥建立协议[24]的安全性。在他们具有开创性的研究中, 模型化了对认证和认证的密钥建立协议的攻击, 设计了几个简单的协议(实体认证和认证密钥协商), 并证明了这些协议是正确的。他们的证明把对协议的所谓成功攻击转化为伪随机性的失败, 即可以用一个多项式时间的分辨器将伪随机函数的输出同真随机函数的输出分辨开来; 换句话说, 否认伪随机函数的存在性。

读者可以回顾我们在 4.7 节中的讨论, 其时给出了假设 4.1 和 4.2。这两个假设是现代密码学的基础。它们表明归约的结果要么是错误的, 要么就是现代密码学基础中的一个重大突破。因为前者更有可能是真实的情况, 所以正如我们所期望的, 归约得到了一个矛盾。

我们只介绍 Bellare 和 Rogaway 的开创性工作中最简单的情形: 对一个基于共享对称密钥的双方实体认证协议[24]的安全性证明。尽管如此, 这种简单的情形也足以让我们看出计算模型证明协议正确性的工作原理。

可证明安全性计算模型中有三个步骤, 我们按照这种步骤介绍 Bellare 和 Rogaway 关于认证协议的开创性工作。在 17.3.1 节中, 我们形式模型化协议参与者和 Malice 的行为。17.3.2 节形式化定义实体互认证的安全目标。在 17.3.3 节我们示范一个证明, 对一个实体互认证协议归约到矛盾。

### 17.3.1 参与者行为的形式模型化

文[24]中考虑的协议是一个双方协议。协议的两个参与者中, 每个都拥有一段可输入输出的有效可执行代码作为协议的一部分。协议由这两部分之间对输入输出值的通信组成。但是, 我们应该注意这里的“通信”应该可以通过 Malice, 而且可能遭到他篡改通信值。

因此, 我们分两步描述一个抽象协议: 一是通过协议参与者拥有的有效可执行函数; 二是通过通信的组合。

#### 17.3.1.1 诚实参与者所拥有协议部分的形式化

形式上, 抽象协议的这一部分由关于下列输入值的一个多项式时间函数  $\Pi$  来描述:

- $1^k$ : 安全参数—— $k \in \mathbb{N}$ (将  $k$  写成一元表示的原因, 请回顾 4.4.6 节和定义 4.7)。
- $i$ : 拥有这一部分协议的主体的身份; 我们将该主体称为“所有者”;  $i \in I$ , 其中  $I$  是共享同一长期密钥的参与者集合。
- $j$ : 所有者意定通信伙伴的身份;  $j \in I$ 。
- $K$ : “所有者”的长期对称密钥(即秘密输入); 在我们基于共享对称密钥的双方协议中,  $K$  也是  $j$  的长期密钥。
- $conv$ : 到目前的对话—— $conv$  是一个比特串; 它随着协议的运行而增长; 新串将级联在它的后面。
- $r$ : 所有者的随机输入——读者可以将  $r$  视为所有者生成的一次性随机数。

因为  $\Pi(1^k, i, j, K, \text{conv}, r)$  的运行时间是关于其输入值大小的多项式(注意  $1^k$  的大小是  $k$ ), 我们可以认为  $K$  和  $r$  的大小是  $k$ , 而  $i, j, \text{conv}$  的大小是关于  $k$  的多项式。

$\Pi(1^k, i, j, K, \text{conv}, r)$  运行一次产生三个值:

- $m$ : 下一个要送的消息—— $m \in \{0, 1\}^* \cup \{\text{“无消息输出”}\}$ ; 它是要在公开网络上发给意定通信伙伴的公开消息。
- $\delta$ : 所有者要做的判决—— $\delta \in \{\text{Accept, Reject, No-decision}\}$ ; 所有者对意定通信伙伴所宣称的身份判定是接受、拒绝或是保持不判; 尽管随时都有可能做出拒绝判定, 通常直到协议结束时才做出接受决定。除了“No-decision”外, 一旦做出某个决定, 结果  $\delta$  就不再改变。
- $\alpha$ : 对所有者得到的秘密输出—— $\alpha \in \{0, 1\}^* \cup \{\text{“没有秘密输出”}\}$ ; 协商的会话密钥作为接受运行的结果, 读者可以认为它是对所有者的秘密输出。

从协议部分的这一形式模型化, 我们知道 Bellare 和 Rogaway 对实体认证协议的模型化使用了以下协议要素: 密码学操作、参与者身份、新鲜性标识符和会话的消息(关于这些协议要素的含义请回顾 11.4 节)。

有时我们用  $\Pi(1^k, I)$  表示参与实体均属于集合  $I$  时的抽象协议。

给定任意一对  $i, j \in I$  (即共享一个长期对称密钥的两个主体) 以及  $s \in \mathbb{N}$ , 我们用  $\Pi'_{i,j}$  表示参与者  $i$  试图在一次会话中认证参与者  $j$ , 并且参与者  $i$  把这次会话标记为  $s$ 。这种尝试可能是由  $i$  发起的, 也可能是对意定通信伙伴  $j$  发送的某个消息的应答。事实上, 我们一般(和惯例上)都把这一尝试看做是对 Malice 的一个随机预言提问的应答。这个一般性是我们通过对通信的形式化得到的。

### 17.3.1.2 通信的形式化

Bellare 和 Rogaway 沿用了 Dolev 和 Yao[102] 的模型(见 2.3 节): 攻击者 Malice 控制整个通信网络。

假设 Malice 具有网络观察能力, 对此我们在第 2 章和第 11 章中已经很熟悉了, 对任意给定的一对  $i, j \in I$  (即他们共享一个长期对称密钥), Malice 可以观察到一系列的  $\Pi'_{i,j}, \Pi'_{j,i}$ , 即使这些函数并不是他自己执行的。但是, 作为一个主动攻击者, Malice 可以做的比被动观察的多得多。他想要在诚实的参与者之间进行多少次会话就可以进行多少次, 而且他可以说服一个参与者(譬如说  $i$ ) 开始一次协议运行, 就好像是在和一个诚实参与者(譬如说  $j$ ) 运行该协议。

因为 Malice 是一个强大的主动攻击者, 我们可以令人信服地假设 Malice 拥有  $\Pi'_{i,j}, \Pi'_{j,i}$  作为黑盒形式的预言机( $i, j \in I, s, t \in \mathbb{N}$ )。这意味着 Malice 可以通过向  $i$  提供输入值( $i, j, s, \text{conv}$ )而向  $\Pi'_{i,j}$  提问, 他也可以类似地向  $\Pi'_{j,i}$  提问。当 Malice 用输入值( $imj, s, \text{conv}$ )向预言机  $\Pi'_{i,j}$  提问时,  $i$  在 Malice 的输入中加入它自己的秘密输入  $K$  和随机输入  $r$ , 所以  $\Pi'(1^k, i, j, K, \text{conv}, r)$  可以执行。执行完毕之后,  $i$  会向外发送一条输出消息  $m$  (如果有的话), 或者“无消息输出”, 以及决定  $\delta$ , 自己保留秘密输出  $\alpha$ 。向外发送的输出结果当然被 Malice 得到, 所以 Malice 可以进一步继续他的攻击。

在 Dolev-Yao 的通信威胁模型下, 预言机在做出“接受”决定之前, 它一直认为收到的所有提问都来自 Malice。



不失一般性,我们不妨假设总是存在一种特别友好的攻击者,我们称这种攻击者为“良性攻击者”,他的行动只限制于选择预言机对  $\Pi'_{i,j}$  和  $\Pi'_{j,i}$ ,然后如实地将从  $\Pi'_{i,j}$  开始的每次预言机回答传送给另一个预言机。换句话说,良性攻击者的第一个提问是  $(inj, s, "")$  (其中“”表示一个空串),生成应答  $m_1^{(i)}$ ;他的第二个提问是  $(j, i, t, m_1^{(i)})$ ,生成应答  $m_1^{(i)}$ ;如此等等,直到两个预言机都得到“接受”判定而终止。因此,良性攻击者就像连接  $i$  和  $j$  的一段导线。后面我们将看到,对于一个可证明安全的协议, Malice 如果希望被攻击的主体输出“接受”判定,那么他的行为就只能限于良性攻击者。

在某一次特殊的协议运行中, Malice 的第  $t$  次提问发生在时间  $\tau = \tau_t \in \mathbb{R}$ 。对于  $t < u$ , 我们要求  $\tau_t < \tau_u$ 。

### 17.3.2 相互认证的目标:匹配对话

Bellare 和 Rogaway 定义了匹配对话(注意这里的对话不止一次)的概念,作为相互认证的安全目标。

预言机  $\Pi'_{i,j}$  的对话是它发送出(和接收到)的一系列时序消息序列,以及因此接收到(和发送出)的应答。令  $\tau_1 < \tau_2 < \dots < \tau_R$  (对某个正整数  $R$ ) 是  $\Pi'_{i,j}$  在其对话时记录的一个时间序列。对话可以由下述序列表示:

$$conw = (\tau_1, m_1, m'_1), (\tau_2, m_2, m'_2), \dots, (\tau_R, m_R, m'_R)$$

这个序列表明在时间  $\tau_1$ , 用  $m_1$  提问预言机  $\Pi'_{i,j}$ , 应答是  $m'_1$ , 然后,在迟些时间  $\tau_2 > \tau_1$ , 用  $m_2$  提问该预言机, 应答是  $m'_2$ ; 一直继续到时间  $\tau_R$ , 用  $m_R$  提问, 应答是  $m'_R$ 。

我们提醒读者在 Dolev-Yao 的通信威胁模型下, 预言机  $\Pi'_{i,j}$  要假设这个对话发生在它和 Malice 之间, 除非在时间  $\tau_R$  它做出了“接受”判定。我们把所有的对话都看做似乎是 Malice 发起的, 这样处理起来非常方便。所以, 如果  $m_1 = ""$ , 那么我们称  $\Pi'_{i,j}$  为发起者预言机, 否则我们称之为应答者预言机。

令

$$conw = (\tau_0, "", m_1), (\tau_2, m'_1, m_2), (\tau_4, m'_2, m_3), \dots, (\tau_{2t-2}, m'_{t-1}, m_t)$$

为预言机  $\Pi'_{i,j}$  的一次对话。如果存在时间序列  $\tau_0 < \tau_1 < \tau_2 < \dots < \tau_R$  和

$$conw' = (\tau_1, m_1, m'_1), (\tau_3, m_2, m'_2), (\tau_5, m_3, m'_3), \dots, (\tau_{2t-1}, m_t, m'_t)$$

其中  $m'_i = \text{“无输出消息”}$ , 我们就称  $\Pi'_{j,i}$  与  $conw$  有一个匹配的对话  $conw'$ , 这两次对话就称为匹配对话。

给定一个协议  $\Pi$ , 如果无论何时  $\Pi'_{i,j}$  和  $\Pi'_{j,i}$  被允许完成一次协议运行(我们再次提醒, 在做出“接受”决定之前, 两个预言机都认为它们是在与 Malice 运行协议), 它们都总有匹配对话, 那么显然 Malice 就不能进行任何比良性攻击者更有危害的攻击, 即他只是像一根导线那样诚实地工作。

现在我们就可以形式化地阐明什么是安全的相互实体认证协议。

**定义 17.1** 我们称  $\Pi(1^k, \{A, B\})$  是  $A$  和  $B$  之间的一个安全的相互认证协议, 如果下列陈述除了一个关于  $k$  的可忽略概率以外都成立: 预言机  $\Pi'_{A,B}$  和  $\Pi'_{B,A}$  都做出“接受”判定当且仅当它们有匹配对话。

当我们使用这个定义证明一个协议安全时,显然匹配对话的存在便意味着两个预言机都做出“接受”,因为预言机一旦完成在(匹配)对话中的那一部分,它就会接受;而相反方向是不平凡的:双方都接受便意味着匹配对话的存在。这样, Malice 攻击协议的目标是在它们没有匹配对话的情况下,使两个预言机都接受。因此下面的定义与我们的关系更大,而且在我们对协议的安全性证明中会用到:

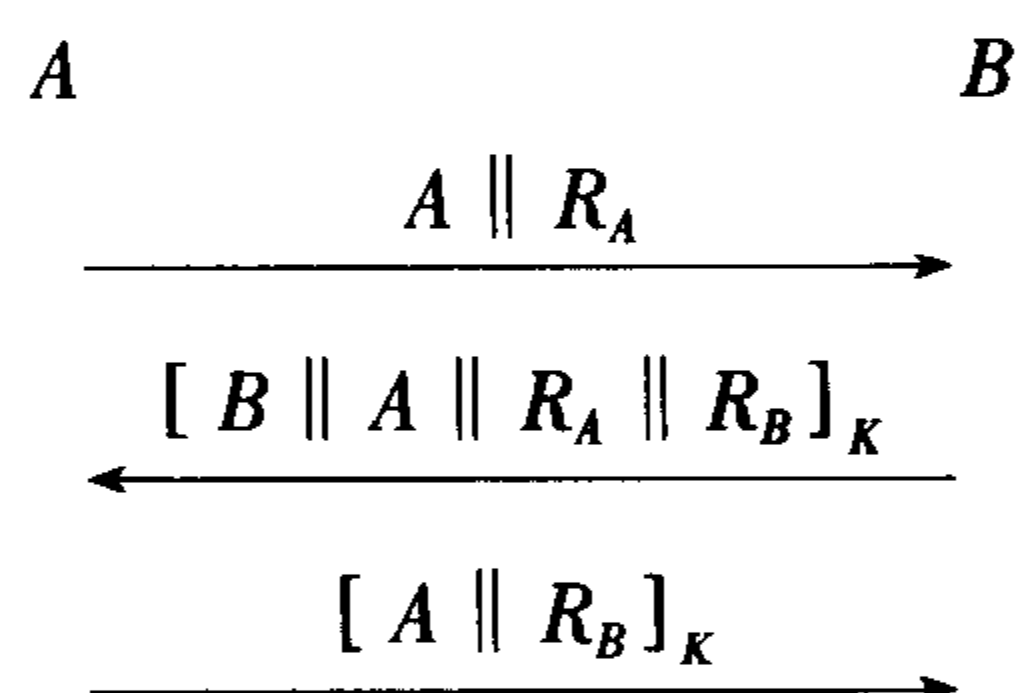
**定义 17.2** 我们称  $\Pi(1^k, \{A, B\})$  是  $A$  和  $B$  之间的一个安全相互认证协议,如果 Malice 不能以一个关于  $k$  的不可忽略概率成功。这里 Malice 的成功定义为  $\Pi'_{A,B}$  和  $\Pi'_{B,A}$  都给出“接受”判定,但它们没有匹配对话。

### 17.3.3 MAP1 协议及其安全性证明

Bellare 和 Rogaway 通过给出一个称为 MAP1 的简单相互实体认证协议并证明其安全性,演示了他们的形式化证明技术。MAP1 在协议 17.6 中具体描述。

#### 协议 17.6 MAP1

**前提:** Alice( $A$ )和 Bob( $B$ )共享一个大小为  $k$  的秘密对称密钥  $K$ ;  $R_A$  是 Alice 的一次性随机数,  $R_B$  是 Bob 的一次性随机数,它们的大小都是  $k$ ;  $[x]_K$  表示对  $(x, \text{prf}_K(x))$ , 其中  $x \in \{0,1\}^*$ ,  $\text{prf}_K: \{0,1\}^* \mapsto \{0,1\}^k$  是一个由密钥  $K$  控制的伪随机函数。



在这个协议中, Alice 以给 Bob 发送  $A \parallel R_A$  开始,  $R_A$  是其长度为  $k$  的一次性随机数。Bob 通过构造一个长度为  $k$  的一次性询问随机数  $R_B$  来应答,并将  $[B \parallel A \parallel R_A \parallel R_B]_K$  发回给 Alice。Alice 检查这个消息是否具有正确的形式以及是否正确地标示了来自 Bob 的消息。如果是, Alice 就发回消息  $[A \parallel R_B]_K$  给 Bob,并接受。Bob 再检查最后的消息是否具有正确的形式以及是否正确地标示了来自 Alice 的消息,如果是则接受。

如果 Alice 和 Bob 都和一个良性攻击者相连,那么由  $\tau_0 < \tau_1 < \tau_2 < \tau_3$ , Alice 会接受下面的对话:

$$\text{conv}_A = (\tau_0, \text{“”}, A \parallel R_A), (\tau_2, [B \parallel A \parallel R_A \parallel R_B]_K, [A \parallel R_B]_K)$$

而且 Bob 会接受下面的对话:

$$\text{conv}_B = (\tau_1, A \parallel R_A, [B \parallel A \parallel R_A \parallel R_B]_K), (\tau_3, [A \parallel R_B]_K, \text{“无输出消息”})$$

显然,  $\text{conv}_A$  和  $\text{conv}_B$  是两个匹配对话。

为了证明在 Alice 和 Bob 与任意类型的多项式有界攻击者(即 Malice)相连的情况下 MAP1 是安全的, Bellare 和 Rogaway 考虑了两个试验。第一个试验中, MAP1 的  $\text{prf}_K$  是一个真随机函

数,即  $MAP1'_{A,B}$  和  $MAP1'_{B,A}$  以某种方式共享了函数  $\text{prf}_k$ ; 当它们给这个函数输入一个给定的  $x$  时,  $\text{prf}_k(x)$  的结果是在  $\{0,1\}^k$  上均匀分布的一个比特串。当然,我们必须承认现实世界中不存在这种共享函数。第二个试验中,和现实情况一样,  $MAP1$  是用一个伪随机函数族实现的。

既然在第一个试验中,  $\text{prf}_K(x)$  是一个均匀分布的  $k$  比特串,那么当  $MAP1'_{B,A}$  看到对话  $\text{conv}_A$  时,它就看到均匀随机串  $[B \parallel A \parallel R_A \parallel R_B]_K$  是用它自己生成的  $R_A$  计算的;所以它可以确信这个不是它的意定伙伴计算(换句话说,是 Malice 计算的)的比特串概率约为  $2^{-k}$ 。不管 Malice 如何行动,即不管是良性的还是恶意的,都是这一数量级的概率值。因此  $MAP1'_{A,B}$  可以得出这样的结论:它的意定伙伴有一个对话以  $(\tau_1 A \parallel R_A, [B \parallel A \parallel R_A \parallel R_B]_K)$  为前缀。这必然向  $MAP1'_{A,B}$  证明了与  $\text{conv}_A$  匹配的对话的存在性,而且这个对话以(关于  $k$  的)压倒性概率已经被它意定伙伴计算过了。

类似地,当  $MAP1'_{B,A}$  看到  $\text{conv}_B$  时,它可以确信它的预期伙伴必然生成了一个与  $\text{conv}_B$  匹配的对话,除了一个  $2^{-k}$  数量级的概率。

所以如果  $MAP1$  用真随机的共享函数,那么 Malice 成功的概率只是(关于  $k$  的)一个可忽略量(关于“Malice 成功”的定义,请回顾定义 17.2)。

证明的剩余部分由矛盾导出。

假设在第二个试验中, Malice 成功的概率是一个(关于  $k$  的)不可忽略量。我们可以构造一个多项式时间的测试  $T$ , 区分随机函数和伪随机函数。  $T$  收到一个函数  $g: \{0,1\}^* \mapsto \{0,1\}^k$ , 该函数根据下列试验选择:

投掷一个硬币  $C$ ;  
if  $C = \text{“HEADS”}$ , 令  $g$  为一个随机函数  
else 随机选择  $K$ , 令  $g = \text{prf}_K$ 。

$T$  的工作就是以某个(关于  $k$  的)不可忽略的优势预测  $C$ 。它的策略是对用  $g$  实现的  $MAP1$  运行 Malice。

如果 Malice 成功(注意  $T$  可以区分 Malice 是否成功,因为  $T$  有两个预言机  $MAP1'_{A,B}$  和  $MAP1'_{B,A}$ , 因此可以看到它们的对话),那么  $T$  预测  $C = \text{“HEADS”}$  (即  $g$  是一个真随机函数), 否则  $T$  预测  $C = \text{“TAILS”}$  (即  $g$  是一个伪随机函数)。这样,我们看到  $T$  区分  $k$  比特输出的随机函数和伪随机函数的优势与 Malice 的成功优势相同,即关于  $k$  的不可忽略量。这与我们通常都认为不存在(关于  $k$  的)多项式时间分辨器可以做这样的函数相矛盾(见假设 4.2)。

在实际中,伪随机函数  $\text{prf}_k$  可以用 CBC 操作模式下的消息认证码实现(CBC-MAC, 见 10.3.3 节),也可以用带密钥的密码学杂凑函数实现(HMAC, 见 10.3.2 节)。这些都是实际中有效的实现方法。

#### 17.3.4 协议正确性计算模型的进一步研究

Bellare 和 Rogaway 在他们开创性的文章[24]中还考虑了认证的会话密钥建立(会话密钥传输)协议(也是两方协议)的正确性。对这些协议,“Malice 成功”意味着或者是在定义 17.2 下成功,或者是成功猜测会话密钥。因为传输的密钥用共享的长期密钥加密,所以成功猜测该密钥与区分伪随机函数和真随机函数有类似的难度。

后来, Bellare 和 Rogaway 将他们最初的工作扩展到了三方的情况:也是简单协议,将一个可信第三方看做是认证服务器[26]。

另外几位研究者进一步研究和开发了类似的方法:例如[38]关于密钥传输协议,[37,39]关于密钥协商协议,[22,58]关于基于口令的协议,[19,67]关于密钥交换协议以及将其设计成正确协议的方法,[68]关于 Internet 密钥交换(IKE)协议。

### 17.3.5 讨论

运用匹配对话这个严格的公式化定义, Bellare 和 Rogaway 给出了协议安全性的一个有用的形式化。我们很快就会看到让参与者获得有意义对话可以很容易地从协议的设计中消除类型缺陷(见 11.7.6 节)以及引起反射攻击(见 11.7.4 节)的一些愚蠢的协议缺陷(设计特性)。用数学方法分析协议也在某种程度上要求协议的设计者和分析者采用正确的或者更精确的密码学服务,从而因误用密码学服务(见 11.7.8 节)而导致的协议缺陷就会更少。

显然,这种证明技术应该与协议设计同时工作。它可以指导协议设计,而且只有这样才能设计正确的协议。

注意,这个方法有一个局限。定义 17.1 以及定义 17.2 没有考虑当两个预言机没有匹配对话时,其中一个预言机接受时的情况。在这种情况下,这些定义不能判定协议是否安全,因为我们只有一个接受。但是,因为接受的预言机确实做出了错误的判定,所以我们也确实应该认为协议不安全。

也许在这种情况下根本不需要注意错误的判定。对于很多安全协议, Malice 总能切断最后一个消息,从而因为预言机得不到最后的消息而使它不能做出接受的判定。显然,其中一个预言机的这种不接受不会使协议不安全。但是,我们应该提醒自己:存在非平凡的认证失败(即不是因为毫无意义的“丢弃最后消息攻击”造成的失败),定义 17.1 和 17.2 不能解决这种类型的失败。我们在攻击 12.1 中演示的 IKE 协议(协议 12.1)认证失败缺陷就是这样的例子。

由于通信的本质,认证协议的可证明安全性比密码学算法的可证明安全性难得多。Bellare 和 Rogaway 的方法开辟了一个正确方向。进一步扩展他们最初有希望的结果是当前一个活跃的研究课题。

## 17.4 正确协议的符号操作观点

正确认证协议的符号操作观点以理论计算机科学家对形式化方法的研究结果为基础。在这种观点下,安全特性被表述成一些可操作的抽象符号的集合,它有时是由一个形式逻辑系统进行的,有时是由一个称为定理证明机的机械工具进行的,最后要得到 YES/NO 的结果。

### 17.4.1 定理证明

定理证明方法可以描述如下:

- 定义一个代数和逻辑公式集,用来描述系统的行为或建立命题,这些命题可以是前提(已知的公式)或者结论(要推导的公式);
- 假定必要的公理集,从而允许使用代数或逻辑方法,由已知公式推导新公式;
- 将要分析的期望系统具有的行为或特性描述成一个待证定理集;
- 利用前提、公理或已证明的定理来证明一个定理以达到希望的结果。

有时,如果运用公理或定理的某些特定规则,定理证明方法中的证明过程就可以机械化。这样的证明工具就称为(机械)定理证明机。应用项重写规则将一个公式重写成范式,就是机械化证明的一个标准范例。例如,众所周知,任何布尔表达式都可以机械化地重写成“合取范式”(CNF)。但是,在大多数情况下,机械定理证明机的证明冗长乏味。还有一个众所周知的现象是机械证明的长度可以是关于要推导公式大小的非多项式有界量(关于非多项式有界量的含义,回顾 4.6 节)。

尽管机械定理证明机容易产生不实际的大证明,但是定理证明的方法可以处理那些行为描述不能用有限结构表示的系统(例如一个有无限状态空间的系统)。归纳证明基于整数的数学命题就是这样的例子。然而,简短的证明通常都要涉及人类的智慧。

基于代数的定理证明方法中,一个必要性质是公理系统必须保持所谓的同余性。它将整数上的同余关系(定义于 4.3.2.5 节)推广到了任意代数结构上。代数结构上的一种二元关系  $R$  称为同余,如果对该结构上的任何二元运算  $\circ$ ,只要

$$R(x, u) \text{ 且 } R(y, v)$$

就有

$$R(x \circ y, u \circ v)$$

同余性也称为代换性或可代换性。当系统具有可代换性时,一个系统元素就可以代换成具有相关行为的另一个元素,根据代换元素之间的关系,系统行为就可以保持不变。如果一个定理证明系统不具备可替换代换性,那么它就不能视为一个健全的系统。因此,可代换性又称为定理证明系统的正确性。无正确性的“定理证明”是没有用的,因为它会产生一个不一致的命题,譬如毫无意义的命题“ $1 = 2$ ”。

定理证明系统的完全性是指其公理系统对于“语义有效性”这个概念的满足状况。基本上,如果一个定理证明系统是完全的,那么任何语义上有效的命题都是可以证明的,也就是说,必然存在一系列的公理应用,表明该命题在语法上是有效的。完全性是我们所希望的,但是机械定理证明机一般都不具备这一点。

定理证明方法的目标与其说是寻找系统中的错误,还不如说是证明某些受欢迎的系统性质。这是因为不良性质通常都不能阐述成定理。尽管如此,用定理证明系统对某条希望具有的性质证明的失败经常会引发一些极具洞察力的想法,进而导致隐藏错误的新发现。

认证协议是极易出错的系统。一步就得到一个安全的协议,然后使用定理证明方法证明它的安全性,一般情况下,这一点是很难做到的,所以定理证明方法不如可以直接找错误的方法有用。

### 17.4.2 一种认证逻辑

对协议正确性概念形式化的最初尝试之一是 Burrows、Abadi 和 Needham 的“认证逻辑”,称为 BAN 逻辑[62]。我们可以认为 BAN 逻辑是采用定理证明方法。它提供一个逻辑公式的集合来模型化协议参与者的基本行动,并且以直观的方式给出了基本协议要素的含义:

- 主体看到(sees)一个消息;
- 主体发出(utters)一个消息;
- 主体相信(believes)一个逻辑命题为真或对一个逻辑命题的真假予以裁决;
- 两个(或多个)主体共享(shares)一个秘密(消息);



- 消息加密;
- 消息的新鲜性;
- 逻辑命题的合取;
- 好的共享密钥:从未被 Malice 发现,而且是新鲜的。

BAN 逻辑的公理系统也是在直观的基础上假定的。例如,下述规则(称为“一次性随机数验证”)精确地达到了消息认证中的新鲜性要求:

$$\frac{P \text{ believes fresh}(X) \wedge P \text{ believes}(Q \text{ said } X)}{P \text{ believes}(Q \text{ believes } X)} \quad (17.4.1)$$

结论中的  $Q \text{ believes } X$  可以解释为主体  $Q$  最近发出了消息  $X$ 。这个公理表明:如果主体  $P$  相信消息  $X$  是新鲜的,也相信  $Q$  说过  $X$ ,那么他可以相信  $Q$  最近说过  $X$ 。

BAN 逻辑中,协议的分析首先陈述一组前提(协议假设),然后“理想化”协议消息,这一过程将协议消息转化成逻辑公式。最后,为了证明一条想要的性质,如一段好密钥的陈述,对这些逻辑公式应用公理。

作为一种定理证明方法,BAN 逻辑不具有可以直接寻找协议缺陷的功能机制。但是,应该注意到我们可以反向推导:从希望的目标出发,应用公理得到一系列必要的前提。因此,BAN 逻辑在揭示协议描述所遗漏的暗含假设方面是非常成功的,为了使所希望目标的命题成立,这些假设是必不可少的。遗漏的假设经常带来协议缺陷的发现。很多认证协议的诸多缺陷在开创性研究[62]中就已经被发现。然而,用这种方式发现缺陷高度依赖于(人类)分析者的经验、洞察力甚至还有运气。

协议的理想化过程可能是一个容易出错的过程。文献中的协议都被典型地描述成主体发送/接收消息的序列。使用 BAN 逻辑要求分析者将协议转化成关于(主体之间传送的)消息的公式,这样才可以应用公理。例如,如果 Trent 发送一条包含密钥  $K_{AB}$  的消息,那么消息发送过程可能需要转化成

$$T \text{ said } A \xleftrightarrow{K_{AB}} B$$

它的意思是对于 Alice 和 Bob 之间的通信,密钥  $K_{AB}$  是一个好密钥。这个协议理想化过程似乎是非常直接的,但是,实际上它是一项非常微妙的工作。Mao 发现 BAN 逻辑为协议理想化提供了一个与上下文无关的过程[185]。例如,我们上面演示的协议理想化步骤没有考虑协议的上下文就完成了。这事实上是一种非常危险的简化。Mao 强调了 BAN 逻辑中协议的理想化必须是一个与上下文有关的过程。

BAN 逻辑最初提出时还有一个缺陷:它缺乏对基本语义的形式化定义,语义是公理系统健全性的基础。因此,作为定理证明系统,逻辑可能会被认为是不健全的;另一个结果是有些公理甚至缺乏有意义的类型。例如在式(17.4.1)中列出的一次性随机数验证规则中,对  $X$ (一次性随机数)的大多数情况, $Q \text{ believes } X$  也包含一个类型错误:即使是在理想化之后, $X$  通常也不是一个逻辑公式(在解释一次性随机数验证规则的含义时我们已经做了必要的纠正)。

[3, 288]尝试为 BAN 逻辑提供基本的语义并论证其健全性。

对于 BAN 逻辑,著名的扩展有 Gong、Needham 和 Yahalom 的 GNY 逻辑[133], van Oorschot 的扩展[294]和 Kailar 的责任逻辑[158]。

GNY 逻辑扩展包括可识别性概念,它的基本思想是具体说明协议消息的类型信息以防止类型错误,还包括主体拥有消息的概念,它是主体生成或识别了这些消息的结果。



van Oorschot 扩展为了便于检验“基于公钥的认证的密钥建立协议”,已经用这个扩展的逻辑分析了基于 Diffie-Hellman 的三个密钥协商协议,其中包括 STS 协议(我们在 11.6.1 节已经检验过协议 11.6)。

在 Kailar 的扩展中,Kailar 令人信服地证明了在电子商务应用中,更重要的是责任而不是信任,并提供了可以表达这些性质的一种句法以及证明这些性质的一组证明规则。与 BAN 逻辑类似,这三种扩展都缺乏对逻辑健全性的形式化语义模型。

尽管如此,BAN 逻辑无疑是一个重要的开创性研究,它激发了对于用形式化方法分析认证协议的研究。

## 17.5 形式化分析技术:状态系统探查

另一种分析复杂系统行为的通用方法是将复杂系统模型化为一个(有限)状态系统。状态系统的性质可以通过一些状态满足关系来表示。系统行为的分析通常涉及到状态空间的探查,用以验证某些性质是否可以满足。这种方法通常被称为模式检验。

通常情况下,模式检验可以是防止某些不良性质并使它们不再出现的方法,也可以是保证某些希望的性质最终会出现的方法。通常认为前一种检验考虑的是系统的安全性,而后一种考虑的则是系统的活跃性。

目标是安全性的检验看上去似乎与应用于认证协议分析的模式检验技术关系更大。

### 17.5.1 模型检验

模型检验方法可以描述如下(在描述时我们将使用一些具体的例子):

- 有限状态系统的操作行为被模型化为一个有限状态转移系统,通过与事件集环境交互,该系统可以发生状态转移;这样的系统称为“标记转移系统”(LTS)。
  - 例如,我们在例 4.1 中给出的单带图灵机 Div3 就是一个 LTS,通过扫描输入纸带上的比特串,它可以发生状态转移。
- LTS 的每个状态都机械地译成(或赋以)一个逻辑公式。
  - 例如,Div3 的每一个状态都可以描述成  $\{0, 1, 2\}$  中的一个适当逻辑命题,这些命题分别声明“目前扫描到的比特串表示一个模 3 同余于 0, 1 或 2 的整数”。
- 作为分析目标的系统性质也明确地译成一个逻辑公式。
  - 例如,对 Div3 的目标陈述可以是“接受的比特串是被 3 整除的整数”。
- 以符号的形式执行 LTS;一次符号执行产生一个“迹”

$$\pi = f_0 e_1 f_1 e_2 \cdots e_n f_n$$

其中  $f_0, \cdots, f_n$  是逻辑公式,  $e_1, \cdots, e_n$  是事件。

— 例如,Div3 接受的所有迹中的所有比特串构成了 DIV3 语言。

- 可以用机械过程检验目标公式是否可以用某个迹中的某个公式满足;这里可满足的含义是该目标公式是某个迹中某个公式的逻辑推理。
  - 例如,对于 Div3,可以机械地检验任何终止迹是否满足目标公式“接受的比特串代表一个可被 3 整除的整数”。

我们要注意,与定理证明的情况不同,定理必须是对希望的系统目标的断言,而在模型检验中,目标公式不仅可以模型化希望系统具有的性质,也可以模型化系统的不良性质。例如,

“Malice 知道新分配的会话密钥  $K$ ”

就是一个公式,它模型化了密钥分配协议的一条不良性质,该公式说明协议有其他主体参与运行。当目标公式模型化一条不良性质时,可满足检验的结果产生一个迹,这个迹明确地描述了一个系统错误。因此,模型检验方法在寻找系统错误的模式中可以工作。

我们应该强调,对于用于分析认证协议的模型检验技术,缺陷的检测是主要的工作模式。

#### 17.5.1.1 模型检验中的系统组合

当我们设计一个复杂系统时,用更简单的组件来构造通常也会更容易一些。

例如,如果我们要设计一个图灵机,接受可以整除 6 的比特串(我们把这个机器记为 Div6),一种简单的方法是设计一个接受偶数的机器(记为 Div2)以及 Div3(在例 4.1 中给出),然后用一个“合取组合”把这两个机器组合起来。设计 Div2 和 Div3 很可能比直接设计 Div6 简单得多。

在这种合取组合中,Div2 和 Div3 扫描它们各自内容相同的输入带,而且这两个机器是并行同步的,也就是说,它们每一步移动都在同一时间。假设这个组合机器接受一个输入当且仅当两个组成机器都接受该输入。显然这种组合方法确实得到了一个正确的 Div6(这种“合取组合”的一个具体实现将在 17.5.3 节中举例说明)。

一个更具说服力的例子是 Div2 和 Div3 的“析取组合”,这将产生一个机器接受被 2 或 3 整除的串,即它接受下面序列中的任何一个数:

0, 2, 3, 4, 6, 8, 9, 10, 12, 14, 15, ...

这里“析取组合”意味着只要 Div2 或 Div3 以接受而停下来,组合得到的机器也将以接受而停下来。显然,如果我们不使用组合方法,那么设计如此笨拙的机器可能也是一项相当令人不舒服的工作。

用模型检验方法在认证协议中查找缺陷也可以通过系统组合的方法得以简化。

事实上,从认证协议中查找缺陷的问题就是一个检查系统的过程,这个系统总是比代表协议描述部分的系统大。协议的具体描述最多只是描述合法参与者应该如何行动。但是,成功的攻击者总是要描述一个更大系统的行为,在这个更大的系统中,Malice 与(一些)合法参与者“和睦相处”(即 Malice 成功欺骗一个参与者或者在不被发现的情况下发现一个秘密)。

因此,在分析认证协议的模型检验方法中,不仅要模型化协议中描述的合法参与者的角色,而且还要模型化 Malice 的一些典型行为(后面我们会看到怎样模型化 Malice 的典型行为)。每个模型化的组成部分都是一个 LTS。它们将组合成一个更大的 LTS 然后来检验。模型检验工具中的组合操作经常模型化了系统组成部分之间的异步通信。这里,“异步”的意思是组合系统可能会因为其中一个组成子系统移动一步而移动一步。这模型化了 Malice 的移动可能独立于诚实协议参与者的情形。

我们在最初介绍模型检验方法时强调过,这种方法适于处理有限状态系统。事实上,模型检验技术只能处理可模型化为有限状态 LTS 的系统。在认证协议的分析中,这个限制要求 Malice 是一个计算有界的主体:不考虑与无界计算能力有关的行为。

模型检验方法经常面临“状态爆炸”的问题:系统对应的大 LTS 状态太多,以至于计算资源不能处理。当要分析的系统是一个大的软件或硬件系统时,这个问题变得尤其严重。这样的系统往往要用一个庞大的状态空间来模型化。幸运的是,在 Malice 计算有界这个合理假设下,大多数认证协议都可以用很小的 LTS 模型化。因此,模型检验技术特别适用于认证协议的分析。

现在,我们简要介绍分析认证协议的两模型检验技术。

### 17.5.2 NRL 协议分析机

Meadows 开发了一种基于 PROLAG 的协议检测工具,称为 NRL 协议分析机[196],“NRL”是美国海军研究实验室(Naval Reserch Laboratory)的缩写。

与分析认证协议的其他方法类似,NRL 协议分析机也基于 Dolev 和 Yao 的通信威胁模型[102](见 2.3 节)。所以 Malice 可以观察网络中的所有消息传输,可以窃听、读取、修改甚至破坏消息,也可以对截获的消息执行变换操作(如加密或者解密,只要他有正确的密钥),还可以假冒某个主体将他的消息发送给其他主体。但是,Malice 的计算能力是多项式有界的,因此存在一个“字”集,在协议运行之初不允许 Malice 知道,而且如果协议是安全的,那么在协议运行完后,他仍然不知道这些字。这个字集可以是秘密的消息,也可以是协议要保护的密码学密钥,我们称这个集中的字为“禁用字”。

NRL 除了是一种模型检验方法以外,还有一些项重写系统的味道。它使用了一个修改的 Dolev-Yao 威胁模型,称为“项重写 Dolev-Yao 模型”。我们可以假设 Malice 在操作一个项重写系统。如果 Malice 的目标是找到一个禁用字,那么证明一个协议安全的问题就变成项重写系统中的一个字问题:禁用字应该仍然是禁止的。等价地,证明一个协议不安全的问题就变成建立一个项重写序列,该序列表明 Malice 可以获得某些“禁用字”。

在 NRL 协议分析机中,把协议模型化为一个“全局”有限状态系统。这个全局状态系统由一些“局部”状态系统和 Malice 的一些状态信息组成。每个局部状态系统描述一个参与协议的诚实主体。构造系统行为的这种方式沿用了由比较容易理解的组件构造复杂系统的标准方法(见 17.5.1.1 节)。

在全局状态系统中,涉及 Malice 的部分模型化了他怎样利用协议的运行生成知识。Malice 的目标是生成一个“禁用字”,这可能是由全局状态系统中涉及 Malice 的那一部分实现的,该部分模型化了 Malice 试图使诚实参与者达到与协议目标函数不相容的某个状态。这样的状态我们称为“不安全状态”。如果一个协议有缺陷,那么不安全状态就应当是可达的。在项重写模型中,达到一个不安全状态等价于建立一个项重写序列,这表明 Malice 可以获得他本不应该获得的某些字(即“禁止字”)。

NRL 协议分析机中,定义了一个状态转移规则集。当某些条件成立时,一个状态转移规则可以被“启用”,而且规则“运行”后会出现某些结果:

在规则可以启用以前:

- 必须赋予 Malice 一些字;
- 相关的局部状态必须和某些值联系起来;

规则运行以后:

- 某个诚实主体要输出一些字(因此被 Malice 获悉);

— 相关的局部状态将和一些新值联系起来。

这些规则中涉及到的字服从一组项重写规则。典型的情况是,有三个规则分别用以获得相同的概念以及加解密是互逆函数的事实。这三个规则是:

$$\begin{aligned} \text{encrypt}(X, \text{decrypt}(X, Y)) &\rightarrow Y \\ \text{decrypt}(X, \text{encrypt}(X, Y)) &\rightarrow Y \\ \text{id\_check}(X, X) &\rightarrow \text{YES} \end{aligned}$$

为了执行分析, NRL 协议分析机的用户提交一个状态描述来询问分析机, 该描述使用 Malice 知道的字(如不安全状态的描述)。然后 NRL 协议分析机进行反向查找, 试图找到全局状态系统的初始状态。这在 PROLOG 中可以很自然地完成, 只要统一项重写规则右边的当前状态, 然后从左边还原以前状态的可能描述即可。如果找到了初始状态, 那么该系统确实不安全; 否则就尝试证明这个不安全状态是不可达的, 这一点是通过证明能达到这一特殊状态的任何状态也都是不可达的。这种查找经常产生一个无限迹, 其中, Malice 为了知道一个字 A, 他必须知道字 B, 而为了知道字 B, 他又必须知道 C, 依此类推。分析机包括某些特性允许用户证明关于状态类不可达的引理。最终的目标是将状态空间缩减为一个足够小的空间, 从而可以用穷举搜索判定对协议的一个攻击是否可能。

我们应该注意 NRL 协议分析机中使用的主要算法可以回答状态是否可达的问题。大家知道这种算法不能确保会终止。因此, 对某些检测程序要限制允许递归调用的次数。使用 NRL 这个工具似乎需要使用者在正确编码协议转移规则, 描述不安全状态方面有相当高的专业技术水平。而且这个工具还有一个固有的局限, 它尤其不适用于密钥建立协议。

NRL 协议分析机用来分析过很多认证协议, 而且成功地发现或者证明了其中某些协议的已知缺陷。包括 Needham-Schroeder 公钥认证协议(协议 2.5)[195](为了分析这个协议, Meadows 比较了使用 NRL 协议分析机的分析和 Lowe 使用模型检验机 FDR 的分析[183])、Simmons 的一个“选择广播协议”[194, 276]、Tatebayashi-Matsuzaki-Newman 协议[162]、Internet 密钥交换协议(IKE, 见 12.2.3 节, 在基于签名的“阶段 2”交换协议中发现了一个反射攻击)[137, 197]以及安全电子交易协议(SET)[261, 198]。

### 17.5.3 CSP 方法

CSP 表示通信时序进程(communicating sequential processes), 它主要是 Hoare 的工作[139]。对其语义进行更新之后, 这个名称被改为 TCSP(Theoretical CSP)[61]。最后在[138]中, TCSP 重新命名为 CSP。

CSP 属于一类称为进程代数的系统。它沿用了构造抽象计算结构的代数方法(见第 5 章)。作为代数, CSP 是关于项的语言, 对这些项定义了运算。这些运算服从“封闭律”, 即 CSP 的项在这些运算下构成一个闭包(回顾第 5 章的定义 5.1、定义 5.12 和定义 5.13)。每个运算对应一个运算语义, 给出所构造的项的含义。很快我们就会看到基本的 CSP 项和运算。

#### 17.5.3.1 行动与事件

在 CSP 中, 根据系统可以执行的行动模型化一个系统。行动是顺序发生事件的一个有限序列, 包括一个 0 长的序列, 它表示“什么都不做”。所有可能事件的集合(在分析一开始就固

定下来)称为进程的字母表,记为 $\Sigma$ 。所以,对于任何行动 $a$ ,都有 $a \in \Sigma^*$ 。例如,对于我们稍后就要给出的几个进程的字母表都是 $\Sigma = \{0,1\}$ ,这些进程可以执行行动的一个实例就是具有某种性质的一个比特串(很快就会明白)。

用 CSP 模型化协议和通信系统时,行动可以是一个原子消息,也可以是一个消息序列。如果 $M$ 和 $N$ 是消息序列,那么 $M.N$ 也是一个消息序列。在不造成费解的情况下,我们可以省略消息序列中的符号“.”。

### 17.5.3.2 进程

进程是系统的组成部分,是用 CSP 描述的实体,对它们的描述是根据它们可以从事的可能行动给出的。图 17.1 列出了最基本的 CSP 进程以及与它们关联的操作语义。

- $STOP$  (“无行动”:什么也不做);
- $a \rightarrow P$  (“前缀”:执行行动 $a$ ,然后像 $P$ 一样工作);
- $P \square Q$  (“确定性选择”:根据环境中发生的外部作用,响应行为或为 $P$ 或为 $Q$ );
- $P \sqcap Q$  (“非确定性选择”:响应行为如 $P$ 或 $Q$ ,没有明确原因,或许因为天气原因);
- $/a$  (“隐蔽”:不用管行动 $a$ );
- $\mu X. P(x)$  (“递归”:重复 $P$ 的行为,其中 $X$ 是一个变量,含义: $\mu X. P(x) \stackrel{\text{def}}{=} P(\mu X. P(x))$ );
- $P \parallel Q$  (“协同性”:当 $P$ 和 $Q$ 都可以执行同一行动时,它们一起展开通信);
- $P \text{ III } Q$  (“交错”:不通信就排列 $P$ 和 $Q$ ,含义:它们没有必要执行同一个行动)。

图 17.1 CSP 语言

图 17.1 中的基本操作是 CSP 进程的基本组成模块,这些进程用来模型化和描述有限状态系统的行为。有了这些组成模块和关联的操作语义,CSP 语言就很强了,足以描述复杂的有限状态系统。

例如,在例 4.1 中给出的图灵机 Div3,它是一个有限状态系统,我们可以用一个 CSP 进程重新简明描述。例 17.1 给出了描述,只使用“前缀”、“确定性选择”和“递归”操作。

#### 例 17.1 Div3 的 CSP 描述

$$\begin{aligned} \text{Div3} &\stackrel{\text{def}}{=} S_0; \\ S_0 &\stackrel{\text{def}}{=} (0 \rightarrow S_0) \square (1 \rightarrow S_1) \square (\{\} \rightarrow STOP); \\ S_1 &\stackrel{\text{def}}{=} (0 \rightarrow S_2) \square (1 \rightarrow S_0); \\ S_2 &\stackrel{\text{def}}{=} (0 \rightarrow S_1) \square (1 \rightarrow S_2) \end{aligned}$$

□

进程 Div3 是由一些子进程以递归的方式定义的。所有这些子进程,除了  $STOP$ ,均对 $\Sigma = \{0,1\}$ 中的事件做出反应<sup>①</sup>;对什么都不做出反应的  $STOP$  意味着终止。易于验证 Div3 有下列特性:

$$\text{对所有 } a \in \text{DIV3} \cup \{\}, \text{Div3} = a \rightarrow STOP$$

其中对于语言 DIV3 的含义,见例 4.1。

① 事实上,回忆我们在 17.5.3.1 节中关于将原子行动 $\{e\}$ 省略成 $e$ 的约定;这些子进程,除了  $STOP$ ,对 $\Sigma^*$ 中的原子行动 $\{0\}$ 和 $\{1\}$ 进行反应; $S_0$ 还对空操作 $\{\}$ 反应,使得 Div3 终止。



### 17.5.3.3 迹

进程  $P$  的语义定义为它可能执行的事件序列的集合,记为  $traces(P)$ 。迹的例子有  $\{\}$  (空迹)和  $1001$  (Div3 的一个可能迹)。

定义在两个迹集合  $T$  和  $T'$  上的操作“.”为:

$$T \cdot T' = \{tr \cdot tr' \mid tr \in T \wedge tr' \in T'\}$$

其中级联序列  $tr \cdot tr'$  已在 17.5.3.1 节中定义。

### 17.5.3.4 进程分析

如果进程  $P$  所有的迹构成的集合是语言 (例如一个说明)  $L$  的一部分,那么就说进程  $P$  满足语言  $L$ :

$$P \text{ sat } L \Leftrightarrow traces(P) \subseteq L$$

进程  $P$  细化进程  $Q$ , 如果  $traces(P) \subseteq traces(Q)$ 。这表明如果  $Q$  满足语言  $L$  (即  $Q \text{ sat } L$ ), 那么  $P$  也满足  $L$ 。

例如, 因为  $traces(\text{Div3}) = \text{DIV3} \cup \{\}$ , 所以  $\text{Div3} \text{ sat } \text{DIV3} \cup \{\}$ 。事实上, Div3 细化一个执行所有比特串的进程。自然地, 我们还有  $\text{STOP}$  细化 Div3 ( $\text{STOP}$  细化任何协议)。我们即将看到细化 Div3 的一个非平凡进程。

模型检验技术允许对有限状态系统机械地检测细化关系, 这要使用一种我们称之为错误分叉细化 (FDR, 它是形式化系统 (欧洲) 有限公司的产品, 关于细节可以访问该公司的站点 <http://www.fsel.com/index.html>) 的工具 [250]。

### 17.5.3.5 CSP 中的系统组合

在 17.5.1.1 节中, 我们论述过系统组合在模型检验系统中起着至关重要的作用。在 CSP 中, 系统组合可以通过应用“协同性”和“交错”操作机械地完成 (这两个操作一起构成了 CSP 系统的组合操作)。

这里我们给出一个例子说明 CSP 组合操作的能力。在 17.5.1.1 节中, 我们建议了一种实现 Div6 的方法, Div6 接受被 6 整除的整数比特串。那种方法将 Div6 看做是 Div2 和 Div3 的联合运行, 当两个子机都接受一个串时, 它就接受这个串。这种方法虽然不复杂, 但是它只是一个抽象的想法, 并没有给出 Div6 的具体构造。

现在, 如果我们在 CSP 中构造了 Div2 和 Div3, 那么就可以通过组合 Div2 和 Div3 的 CSP 描述机械地构造 Div6, 得到 Div6 的一个具体 CSP 描述。首先, Div2 非常简单, 我们可以直接构造, 这在下面的例 17.2 中给出。

#### 例 17.2 Div2 的 CSP 描述

$$\begin{aligned} \text{Div2} &\stackrel{\text{def}}{=} R_0; \\ R_0 &\stackrel{\text{def}}{=} (0 \rightarrow S_0) \square (1 \rightarrow R_1) \square (\{\} \rightarrow \text{STOP}); \\ S_1 &\stackrel{\text{def}}{=} (0 \rightarrow R_0) \square (1 \rightarrow R_1). \end{aligned}$$

□



现在应用 CSP 的“协同”操作就得到的 Div6 的机械组合,在例 17.3 中给出。

### 例 17.3 Div6 的 CSP 描述

$$\begin{aligned}
 \text{Div6} &\stackrel{\text{def}}{=} \text{Div2} \parallel \text{Div3} = R_0 \parallel S_0; \\
 R_0 \parallel S_0 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_0) \square (1 \rightarrow R_1 \parallel S_1) \square (\{\} \rightarrow \text{STOP} \parallel \text{STOP}); \\
 R_1 \parallel S_1 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_2) \square (1 \rightarrow R_1 \parallel S_0); \\
 R_0 \parallel S_2 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_1) \square (1 \rightarrow R_1 \parallel S_2); \\
 R_1 \parallel S_0 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_0) \square (1 \rightarrow R_1 \parallel S_1); \\
 R_0 \parallel S_1 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_2) \square (1 \rightarrow R_1 \parallel S_0); \\
 R_1 \parallel S_2 &\stackrel{\text{def}}{=} (0 \rightarrow R_0 \parallel S_1) \square (1 \rightarrow R_1 \parallel S_2);
 \end{aligned}$$

$$\text{STOP} \parallel \text{STOP} = \text{STOP}.$$

□

在例 17.3 中,

$$\text{STOP} \parallel \text{STOP} = \text{STOP}$$

是一个 CSP 公理,称为“吸收律”(关于  $\parallel$  的吸收),它显然为真,因为等号两边什么都不做。

Div6 的机械组合形式确实正确地实现了这个机器(读者可以用几个数值实例检验它,尽管这样的检验不能构成正确性的一个证明)。机械模型检验机 FDR 可以证实 Div6 细化 Div3,而且它也细化 Div2。这两个证实就构成了关于 Div6 确实是目标机器的一个正确实现的证明。

我们还可以由 Div2 和 Div3 机械地构造另一个机器,接受被 2 或 3 整除的整数比特串。这个机械组合可以通过以下两步来实现:i) 应用“交错”操作,进行组合中的两个项能执行同一行动就将“交错”换成“协同”;ii) 应用下述 CSP 中的“死锁公理”:

$$\text{STOP} \sqcap P = \text{STOP} \parallel P = P \sqcap \text{STOP} = P \parallel \text{STOP} = \text{STOP}$$

所得到的机器相当庞大(会有很多状态),这里我们就不确切给出它的描述了。

### 17.5.3.6 安全性协议的分析

CSP 中的组合操作很有用,它使得 CSP 特别适用于模型化以及描述协同行为和通信系统。正是 CSP 的这种特性鼓舞了很多研究者论证它是否适用于认证协议的形式化分析[251, 255, 252]。而且,还有一种模型检验机工具 FDR[250]专门用于检验 CSP 进程的细化关系。Lowe 应用 FDR 模型检验机成功地发现了 Needham-Schroeder 公钥认证协议中一个以前未知的错误[183]。

当分析消息保密特性时,“继承”关系  $I \vdash m$  表示怎样从可得的信息推导新信息。图 17.2 描述了信息推导的继承公理

例如,我们有

$$(\{\{K_1\}_{K_2}\}_{K_3}, K_3) \vdash K_3$$

和

$$(\{\{K_1\}_{K_2}\}_{K_3}, K_3) \vdash \{K_1\}_{K_2}$$

但是没有

$$(\{\{K_1\}_{K_2}\}_{K_3}, K_3) \vdash K_1 \quad (\text{假})$$

继承公理直觉上模型化了 Malice 如何推导信息。当 Malice 试图挫败协议的(保密性)目标时,他可以使用一个他所拥有的初始信息集以及一些发送到网络中的协议消息。根据信息推导的方式,  $I$  原则上是一个无限集。但是,在实际中,给定一个协议,我们总可以将  $I$  限定在“感兴趣”信息的一个有限集上。而且,研究者利用了这样一个事实:没有必要真正构造  $I$ ,只要对有限数量的消息检验  $m \in I$  就可以了。

假设  $I$  是可得信息的初始集合,那么

- If  $m \in I$  then  $I \vdash m$
- If  $I \vdash m$  and  $I \subseteq I'$  then  $I' \vdash m$
- If  $I \vdash m_1$  and  $I \vdash m_2$  then  $I \vdash (m_1, m_2)$  (paring)
- If  $I \vdash (m_1, m_2)$  then  $I \vdash m_1$  and  $I \vdash m_2$  (projection)
- If  $I \vdash m$  and  $I \vdash K \in \mathcal{K}$  then  $I \vdash \{m\}_K$  (encryption)
- If  $I \vdash \{m\}_K$  and  $I \vdash K^{-1} \in \mathcal{K}$  then  $I \vdash m$  (decryption)

图 17.2 CSP 继承公理

总之,在 CSP 方法中,使用机械工具是分析系统行为的一个重要组成部分。机械工具应用一组直观定义的规则。例如,在系统构造中,组合工具可以应用图 17.1 中给出的语义规则通过组合较小系统来构造较大系统;在进程细化检验中,工具可以应用细化的定义(见 17.5.3.4 节)检验迹关系;还有,在信息推导中,工具可以应用图 17.2 中的继承公理。

CSP 的句法,虽然用机械检验机处理它时不会引起任何麻烦,但是,特别是由于它涉及系统各个组成部分之间的通信这一事实,很不适于人们的理解(我们应该注意到,在例 17.3 中我们使用 CSP 的组合操作对 Div6 的机械构造是很简洁的,因为构造过程没有涉及 Div2 和 Div3 之间的任何通信;这两个组成部分都协同地与环境通信,而环境是不需要具体描述的)。

有些认证协议涉及到几个参与者之间的通信,对于本书的大多数读者来说,因为不是形式化方法领域的专家,认证协议的 CSP 模型远远不是可以直接给出的。感兴趣的读者可以参阅 Ryan 和 Schneider 的书[252]。

## 17.6 调和安全性形式化技术的两种观点

从第 14 章开始到现在,我们看到了形式化推理安全性的两种完全不同的观点。

一种观点称为计算观点,我们在第 14 章介绍过,而且在 17.3 节中又再次考虑,它的基础是一个详细的计算模型。以加密的安全性(即保密性)为例,计算观点将保密性看做是区分明文的困难性,也就是说,给定两个明文和它们之一对应的一条密文,攻击者不能判定哪一个明文被加密。安全性推理通常是通过构造一个“归约为矛盾”类型的证明得到的,这里的“矛盾”是计算复杂性领域中人们广泛相信的一个困难问题的有效解。

另一种观点称为符号观点,我们在 17.4 节和 17.5 节中介绍过,它的基础是简单而有效的形式化语言方法。我们再次以加密的安全性为例,符号观点将保密性看做是机械地应用图 17.2 中的继承公理寻找明文的困难性。对公理的机械应用可以基于定理证明技术或者基于状态探查技术。

这两种观点来自两个几乎完全分离的团体。人们注意到,长期以来这两个观点之间,事实上是这两个团体之间,确实有一种令人不愉快的分歧。符号观点,如我们所愿,很简单,但是它有时会因为过于简单而误导研究者得到错误的看法。例如,在符号观点下,有时我们看到的情况是同一对公私钥下签名和加密“互相抵消”,而事实上,即使是教科书式版本,也只有很少的公钥密码算法会这样。更多情况下,从符号观点看加密,它是一个确定性函数,这很容易误导安全工程师们应用一些教科书式加密算法实现加密。

Abadi 和 Rogaway 尽心地开始了一项消除分歧的工作[2]。他们认为结合符号观点和计算观点最终会使彼此都受益。他们详细论述道:

- 这种结合会加强形式化密码学的基础,而且有助于阐明形式化方法中隐含的假设和分歧。通过对实现密码学操作给出明确的要求,它们还可以证实和改善协议的形式化证明与该协议具体实例之间的相关性。
- 随着计算密码学处理日益复杂的系统,高级推理方法似乎是必需的。符号方法蕴涵着这种高级推理的原理,甚至还允许自动化证明。另外,有些符号方法具有自然而强有力的密码学直觉;与这些直觉联系起来能增加计算密码学对人们的吸引力,提高人们对计算密码学的理解力。

Abadi 和 Rogaway 最初沟通分歧的工作是对加密的符号处理给出了一个计算证明。他们的基本思想是证明这两种观点“几乎同态”。首先,在计算观点下,两条不可区分的密文视为等价的。其次,在符号观点下,用继承公理不能被看做有意义的两条密文,也视为等价的。这样,安全性的计算观点就可以视为支撑符号观点安全性的一个良好的形式化基础。

Abadi 和 Rogaway 最初沟通分歧的工作还为进一步研究其他密码学原型安全性形式化处理提供了深刻见解,这些密码学原型包括签名、杂凑函数、认证以及认证的密钥分配协议等。

## 17.7 本章小结

本章我们又回到了认证协议这个实际中很重要的课题,但我们研究的重点是关于这些协议正确性的形式化方法。

首先,我们指出需要一种细化的方法描述协议。我们认识到广泛使用的协议描述方法不很精确,这正是认证协议中误用密码学服务这个常见陷阱的罪魁祸首。接着我们提出了一种细化描述方法,并给出一些细化描述的协议来说明该方法非常有效。

然后我们介绍了协议的形式化分析方法,它们包括在计算观点下证明协议正确性的方法和在符号操作观点下对协议错误的模式检验方法。

因为两种观点都有各自的优势和局限性,所以我们讨论了最近的一个进展,它试图在这两种观点之间寻找关系并调和冲突。

认证协议的形式化分析还是一个处于早期研究和探索的课题。不可避免地,本章给出的内容也有这个特点。

## 习题

- 17.1 密码分组链接操作模式(CBC)广泛用于分组密码以生成随机化的密文。那么在“适于应用”的安全性概念下,这种模式是否可以供保密性服务?

提示: CBC 操作模式是否可以抵抗主动攻击?

17.2 在第 2 章我们给出了一系列对认证协议先攻击后改进的步骤。那么这些过程最后是否得到了安全的协议? 如果没有, 为什么?

17.3 用细化的协议描述方法重新描述 Kerberos 交换协议的各种消息交换(12.4.2 节中)。给出“认证者”的正确描述。

提示: 复习 12.4.3 节。

17.4 误用密码学服务是认证协议和认证的密钥交换协议设计中的一种普遍错误。那么种误用最普遍的形式是什么?

17.5 证明认证协议正确性的 Bellare-Rogaway 模型是一种基于“归约为矛盾”的方法。那么这种归约导致的“矛盾”是什么?

17.6 证明认证协议正确性的 Bellare-Rogaway 模型是否可以应用于任意的认证协议?

17.7 应用 CSP“交错组合”构造一个 CSP 进程, 接受可以被 2 或 3 整除的整数比特串。

提示: 复习 17.5.3.5 节; 要得到的进程相当庞大, 应该使用 CSP 工具做这项工作。



## 第六部分 密码学协议

目前,尤其通过基于 WWW 的工具,在因特网上正在出现和提供越来越多的商务活动、商业交易和政府服务。许多这种应用都要求安全服务。购物、账目、银行业、工作或学籍申请的管理以及税务评估就是其中的少数例子。对于这些应用,认证、保密和不可否认性是最常见的必要安全服务。这些服务用一些简单的密码协议诸如 TLS(SSL,我们在第 12 章已经介绍)就可以提供。

但是,还有一些“更奇特的行为”也可以在线进行,但用诸如 TLS 这样简单的协议不能满足它们需要的安全服务。例如,微支付(如何保证低开销的交易)、电子现金(如何提供花费者的匿名性并防止敲诈)、拍卖(如何分离出获胜标书而不打开标书)、投票(如何保证投票者的匿名性并对勾结免疫)、公平交换(如何保持公平性而不管参与者资源的差别)、时戳与公证(即使其基础机制如一种签名方案在未来被破解了,如何继续保持某种合法的绑定关系)以及定时密钥恢复(是一种刚好  $t$  次乘积后就可恢复的秘密)。

“更奇特”的安全服务通常由“更奇特”的密码协议工具提供。本部分包括两章,在第 18 章我们介绍一种称为零知识协议的密码协议,它构成了“更奇特”的服务底层的核心技术:证明声称的某个性质而不泄漏秘密。在第 19 章,通过提供本书第一个“电话掷币”协议(协议 1.1)的一种具体实现,我们结束本书。该实现提供了一种安全的解决方案进行远程掷币,来生成一个彼此强烈信任的随机比特串,并且该解决方案也是实用的,这在于它使用的是可广泛获取的密码技术,并具有和通常使用的公钥密码相近的效率。



## 第 18 章 零知识协议

### 18.1 引言

两方交互式游戏是密码学的一个基本问题。其中,一方(示证者)向另一方(验证者)证明一个命题成立,但不让后者知道前者是如何证明的。由于缺少示证者所知道的某些信息,验证者不能独自验证命题。这种游戏有一个通用的名称,称为交互式证明(IP)协议(系统)。我们可以将 IP 协议进行的证明看做“在黑暗中证明”。“在黑暗中”一词有两层含义。首先,验证者在确信证明内容的有效性之后,并不能获得示证者为了进行证明所拥有的知识;其次,协议结束后,任何第三方都不可能明白示证者和验证者之间发生过什么有意义的事情。

大胆设想一下,需要“在黑暗中证明”的命题可能是数学中的一个著名公开问题(如哥德巴赫猜想<sup>①</sup>)的肯定答案。示证者也许会担心,如果把知道如何证明的技术“公开”出示给验证者的话,后者可能会盗窃或剽取他/她的知识。在这种情况下,示证者要能够“在黑暗中”进行证明,不提供关于如何证明的任何额外信息就能使后者(现在是 IP 协议中的验证者)确信是肯定的答案。

在许多实际应用中,“在黑暗中”进行证明有很多重要的原因。身份证明作为一种认证手段是 IP 协议的常见应用。与传统的认证方式如由主体发布一个数字签名不同,这里示证者作为被认证的主体,不想让除了意定的验证者以外的任何第三方看到用于认证的通信副本,因此,认证必须“在黑暗中”进行。使用 IP 协议的另外一种常见的例子是证明一部分隐藏信息具有一定的结构。这在一些安全应用中(如在拍卖应用中)是必要的,为了实现别致的服务(如不打开密封标书 $\mathcal{E}_k(x)$ 和 $\mathcal{E}_k(y)$ 而证明 $x > y$ ),其中隐藏的数字(投标)必须在一个有效范围内。

对于 IP 协议,总是关心两个重要问题:

**问题 I** 在交互式证明过程中,验证者获得了多少信息?

**问题 II** 为了让验证者确信,示证者需要多少次交互?

问题 I 的理想答案是未获得任何信息,或者说信息量为 0。具有这一性质的 IP 协议称为零知识(ZK)协议。问题 II 不仅在 IP 协议的实际应用中有重要意义,而且在计算复杂性理论中也有重要意义,因为对于给定的一类问题,该问题的定量解答意味着找到了复杂性一个新的下界。

本章将学习 ZK 协议。我们的学习主要是系统介绍这一主题(包括回答上述两个问题)的各种概念。这些概念非常重要,但是,其中许多都是以多年研究论文积累确定下来的背景资料为基础,而在绝大多数密码学教材中并没有这些内容。为了清晰地理解,在介绍的时候,我们将以许多具体的协议为例来阐明这些概念。相信以这种方式学习 ZK 协议会更容易理解这一主题。

---

<sup>①</sup> 所有大于 2 的偶整数都可以表示为两个素数之和。

### 18.1.1 本章纲要

18.2节介绍 IP 系统的基本概念。18.3节介绍 IP 协议的各种 ZK 特性。18.4节区别 ZK 证明和 ZK 论据的概念。18.5节学习双边差错协议的差错概率特性。18.6节学习轮效率问题。最后,18.7节介绍非交互式 ZK 协议的概念。

## 18.2 基本定义

零知识协议不只是在应用密码学中有重要的应用价值,在 IP 协议框架中,这一主题已经发展为计算复杂性理论的一个重要分支。它包含很多定义,我们将只介绍与应用密码学相关的一些概念。

### 18.2.1 计算模型

现在我们先暂时不管问题 I 和 II,也不考虑信息泄漏和实际效率问题。

我们现在介绍 Goldwasser、Micali 和 Rackoff 定义的交互式证明系统的计算模型[128]。交互式证明协议的基本模型可以表示为  $(P, V)$ , 其中  $P$  是示证者,  $V$  是验证者。在一般情况下, 协议  $(P, V)$  用于证明一种语言的成员归属命题, 其中语言在  $\{0, 1\}^*$  上。18.2.2 节将给出该语言的一般含义, 18.2.3 节将它限制到密码学感兴趣的特殊含义上。

设  $L$  是  $\{0, 1\}^*$  上的一种语言, 对一个成员归属实例  $x \in L$ ,  $P$  和  $V$  必须共享输入  $x$ , 因此它称为公共输入。证明实例表示为  $(P, V)(x)$ 。双方由一个通信信道联系, 通过它进行交互, 交换一系列信息, 表示成:

$$a_1, b_1, a_2, b_2, \dots, a_\ell, b_\ell \quad (18.2.1)$$

这一系列信息交换称为证明副本。证明副本包括  $P$  传输的数据, 称为示证者的副本; 还包括  $V$  传输的数据, 称为验证者的副本。这里, 不仅证明副本的长度  $\ell$ , 而且副本中交换的每一元素的长度, 即  $|a_i|, |b_i|$  ( $i = 1, 2, \dots, \ell$ ), 都关于  $|x|$  的一个多项式有界。证明实例  $(P, V)(x)$  必须在关于  $|x|$  的多项式时间内终止。

一旦协议在关于  $|x|$  的一个多项式时间界内完成时, 输出应当是下列类型:

$$(P, V)(x) \in \{\text{Accept}, \text{Reject}\}$$

这两个值分别表示  $V$  对  $P$  声称的  $x \in L$  拒绝或接受。由于  $(P, V)$  是一个概率系统, 对每一个  $x$ , 输出值  $(P, V)(x)$  是关于这个公共输入  $x$ 、 $P$  的秘密输入值、 $P$  和  $V$  的一些随机输入值的一个随机变量。而且, 证明副本(18.2.1)中的元素也是这样的随机变量。

因为  $(P, V)$  是一个两方游戏, 自然每一方都希望获得比授权更多的优势。一方面, 示证者  $P$  有兴趣使  $(P, V)(x) = \text{Accept}$ , 即使在  $x \notin L$  时也尽可能成立。有这种行为(策略)的示证者称为欺骗示证者, 通常表示为  $\tilde{P}$ 。另一方面, 验证者  $V$  有兴趣发现  $P$  交互中的秘密输入的有关信息。有这种行为(策略)的验证者称为不诚实验证者, 通常表示为  $\tilde{V}$ 。

### 18.2.2 交互式证明协议的形式化定义

现在将给出交互式证明系统的形式化定义。

定义 18.1 设  $L$  是  $\{0,1\}^*$  上的语言。IP 协议  $(P, V)$  称为  $L$  的一个交互式证明系统, 如果

$$\text{Prob}[(P, V)(x) = \text{Accept} \mid x \in L] \geq \epsilon \quad (18.2.2)$$

且

$$\text{Prob}[(\tilde{P}, V)(x) = \text{Accept} \mid x \notin L] \leq \delta \quad (18.2.3)$$

其中  $\epsilon$  和  $\delta$  是常数, 满足

$$\epsilon \in \left(\frac{1}{2}, 1\right], \quad \delta \in \left[0, \frac{1}{2}\right) \quad (18.2.4)$$

概率空间是  $(P, V)$  的所有输入值和  $P$  及  $V$  的所有随机输入值。

概率表达式 (18.2.2) 刻画了  $(P, V)$  的完全性概念的特征, 概率界  $\epsilon$  称为  $(P, V)$  的完全性概率。这表示如果  $x \in L$ , 那么  $V$  将至少以概率  $\delta$  接受证明。

概率表达式 (18.2.3) 刻画了  $(\tilde{P}, V)$  的正确性概念的特征, 概率界  $\delta$  称为  $(\tilde{P}, V)$  的正确性概率。这表示如果  $x \notin L$ , 那么  $V$  将至多以概率  $\delta$  接受证明。

比较定义 18.1 和定义 4.5 (见 4.4 节), 其中复杂性类  $\mathcal{PP}$  的差错概率特征为式 (4.4.1)、式 (4.4.2) 和式 (4.4.3), 我们得到以下结论:

定理 18.1  $IP = PP$ , 其中  $IP$  是其成员归属问题可以由 IP 协议回答的所有语言。  $\square$

而且根据 4.4.1 节中的学习, 我们知道, 通过连续独立重复执行  $(P, V)$  (关于公共输入长度的) 多项式次,  $V$  采用“多数判别准则”做出接受/拒绝决定, 完全性 (或正确性) 概率界可以放大 (或缩小) 到任意接近于 1 (或 0)。

现在回顾一下到目前为止所介绍的概念, 看一个具体的 IP 协议例子: 协议 18.1。

例 18.1 在协议 18.1 中, Alice 是示证者, Bob 是验证者。  $(\text{Alice}, \text{Bob})$  的公共输入是  $X = f(z)$ , 其中  $f$  是协议 18.1 中描述的  $\mathbb{Z}_n$  上的一个单向同态函数。 Alice 的成员归属声明是  $X(\{f(x) \mid x \in \mathbb{Z}_n\})$ 。这实际上是子群的关系  $X \in \langle f(1) \rangle$ , 因为  $X = f(1)^z$  (这个问题对 Bob 困难的一般性条件见注释 18.1)。 Alice 的秘密输入是单向同态函数  $f$  下  $X$  的原像  $z \in \mathbb{Z}_n$ 。

双方在协议中交互  $m$  次, 生成下列证明副本:

$$\text{Commit}_1, \text{Challenge}_1, \text{Response}_1, \dots, \text{Commit}_m, \text{Challenge}_m, \text{Response}_m$$

如果通过了 Bob 进行的所有检验, 协议输出 Accept; 否则输出 Reject。

该协议是完满的, 也就是说, 如果 Alice 确实拥有  $z$  的原像并且遵守协议指令, 那么 Bob 都将接受。

### 完全性

事实上, 完全性表达式 (18.2.2) 以  $\epsilon = 1$  满足, 因为对 Bob 随机选取  $\text{Challenge} \in_v \{0, 1\}$  的所有情况, Alice 的应答都满足他的每一步验证:

$$f(\text{Response}) = \begin{cases} \text{Commit} & \text{如果 } \text{Challenge} = 0 \\ \text{Commit } X & \text{如果 } \text{Challenge} = 1 \end{cases}$$

该协议是稳妥的。

## 协议 18.1 子群成员资格的一个交互式证明协议

( \* 关于协议命名见注释 18.1 \* )

公共输入:

i)  $f: \mathbb{Z}_n$  上的一个单向函数, 满足同态条件:

$$\forall x, y \in \mathbb{Z}_n: f(x + y) = f(x) \cdot f(y);$$

ii) 对某个  $z \in \mathbb{Z}_n, X = f(z)$ ;

Alice 的秘密输入:  $z < n$ ;

对 Bob 的输出: 成员资格  $X \in \langle f(1) \rangle$ , 即  $X$  由  $f(1)$  生成。

重复下列步骤  $m$  次:

1. Alice 选取  $k \in_U \mathbb{Z}_n$ , 计算  $\text{Commit} \leftarrow f(k)$ , 发送 Commit 给 Bob;

2. Bob 选取并发送  $\text{Challenge} \in_U \{0, 1\}$  给 Alice;

3. Alice 计算  $\text{Response} \leftarrow \begin{cases} k & \text{如果 Challenge} = 0 \\ k + z \pmod n & \text{如果 Challenge} = 1 \end{cases}$ , 发送 Response 给 Bob;

4. Bob 检验  $f(\text{Response}) = \begin{cases} \text{Commit} & \text{如果 Challenge} = 0 \\ \text{Commit} \cdot X & \text{如果 Challenge} = 1 \end{cases}$

如果检验显示错误, 他拒绝并中止协议;

Bob 接受证明。

### 正确性

我们需要确定正确性概率  $\delta$ 。

Bob 的检验(第 4 步)依赖于 Alice 发送 Commit 后他随机选取的 Challenge。一致通过 Bob 的验证, 表明下面两种情形:

情形 Challenge = 0: Bob 确信 Alice 知道  $\text{pre-image}(\text{Commit})$ ;

情形 Challenge = 1: Bob 确信  $\text{pre-image}(X) = \text{Response} - \text{pre-image}(\text{Commit}) \pmod n$ 。

由于 Alice 在发送出承诺后不能预测 Bob 随机选择的提问比特, 对于 Challenge = 1 的情形, 她也必须知道  $\text{pre-image}(\text{Commit})$ , 因此也必须知道  $\text{pre-image}(X)$ 。

如果 Alice 不知道  $\text{pre-image}(X)$ , 那么要进行欺骗, 她必须在发送承诺之前猜测提问比特。在欺骗性“证明”中, 她可以按如下方式计算承诺:

● 随机选取  $\text{Response} \in_U \mathbb{Z}_n$ ;

● 猜测 Challenge;

● 计算“承诺”:  $\text{Commit} \leftarrow \begin{cases} f(\text{Response}) & \text{如果 Challenge} = 0 \\ f(\text{Response})/X & \text{如果 Challenge} = 1 \end{cases}$

显然在这一欺骗性“证明”中, Bob 有  $1/2$  的可能拒绝重复的任何一次交互。因此我们得到正确性差错(即 Alice 侥幸欺骗成功)概率为  $\delta = 1/2$ 。如果重复  $m$  次都没有被拒绝, 那么 Alice 成功欺骗的概率以  $2^{-m}$  为界。如果  $m$  足够大, 即  $2^{-m}$  足够小, Bob 就可以充分相信 Alice 不可

能侥幸欺骗成功。例如  $m = 100$ , 可以为 Bob 提供足够高的可信度防止 Alice 欺骗。因此, 一旦 Bob 接受, Alice 的证明就是有效的。

随后(在 18.3.1 节与例 18.2 中)将进一步学习完备零知识的一个性质: 如果函数  $f$  确实是单向的, 那么多项式有界的验证者 Bob 不能获得关于 Alice 秘密输入的任何信息。□

**注释 18.1** 根据同态性, 对所有  $x \in \mathbb{Z}_n$ , 有  $f(x) = f(1)^x$ 。因此协议 18.1 也称为 Alice 证明她拥有  $X$  关于底数  $f(1)$  的离散对数的协议。我们选择称这个协议为“子群成员归属证明”, 因为成员归属问题是 IP 协议解决的更一般的问题。当采用这个(更一般也更合适的)命名时, 我们强调  $\text{ord}[f(1)]$  是  $n$  的某个秘密因子这一更一般的情形, 即  $f(1)$  并不一定生成含有  $n$  个元素的群。对于这种一般情形, 没有 Alice 的帮助, Bob 不能直接验证子群关系。□

注释 18.1 实际上是说, 确定子群成员归属在一般情况下是一个困难问题。我们需要给出关于该困难性更加详尽的说明。注意到尽管集合

$$L_n = \{f(x) = f(1)^x \mid x \in \mathbb{Z}_n\}$$

是一个循环群(因为它由  $f(1)$  生成, 见 5.2.3 节), Bob 确定  $\#L_n \stackrel{?}{=} n$  并不容易。为了回答该问题(即弄清楚  $f(1)$  是 1 的一个本原根还是一个  $n$  次方根, 见 5.4.4 节中的定理 5.11), 需要对  $n$  素分解。只有在  $\#L_n = n$  的情况下, Bob 才可以不必和 Alice 运行协议就能够对协议 18.1 中的子群成员归属问题回答 YES(因为在这种情况下,  $f(1)$  必然生成  $L_n$  的所有  $n$  个元素)。因此, 确定子群成员归属的困难性取决于分解一定规模的  $n$ 。所以, 对于解决子群成员归属问题的协议 18.1, 整数  $n$  必须是一个足够大的合数。由于这个原因, 我们规定  $\log n$  为协议 18.1 的安全参数。

在 18.3.1.1 节我们将看到公共输入参数设置的一种特殊情况, 那样协议 18.1 就退化为证明拥有离散对数的特殊情形。

### 18.2.3 一个复杂性理论结果

读者可以跳过这里(18.2.3 节范围内)给出的内容, 这不会对理解本章其余部分介绍的 ZK 协议概念造成任何困难。

现在来推导计算复杂性理论中的一个事实, 该事实陈述在式(4.5.1)中。在第 4 章我们还不能给出该事实的证明, 现在可以了。

在应用密码学中, 我们不只对回答 IP 子类语言中的成员归属问题的 IP 协议感兴趣。对子类中的任意  $L$ , 成员归属问题  $x \stackrel{?}{\in} L$  具有下面两个特征:

- i) 不知道是否存在确定的或概率的(关于  $|x|$  的)多项式时间算法回答该问题。否则  $P$  在  $(P, V)$  中没有作用, 因为  $V$  自己就能回答该问题。
- ii) 如果一个(关于  $|x|$  的)多项式时间算法拥有问题的一个论据, 那么该问题可以由这个算法回答。

回忆一下对复杂性类  $\mathcal{NP}$  的分类(见 4.5 节), 我们可以知道(i)和(ii)刻画了  $\mathcal{NP}$  类的特征。确切地说, 它们刻画了具有稀疏论据的 NP 问题的特征。由于  $\mathcal{IP} = \mathcal{PP}$ (见定理 18.1), 我们有:

$$NP \subseteq PP$$

因此对任意语言  $L \in NP$ , 存在关于  $L$  的一个 IP 协议  $(P, V)$ , 也就是说, 对任意  $x \in L$ ,  $(P, V)(x) = \text{Accept}$  在关于  $|x|$  的多项式时间内终止。

事实上, 已经有几位研究者用构造方法证明了这一性质。他们对一些 NPC 语言(见 4.5.1 节中的定义 4.11)构造了 ZK(IP)协议, 例如, Goldreich、Micali 和 Wigderson 的用图的三色问题 [126], 以及 Chaum 的布尔表达式的可满足性问题所构造的 ZK 协议 [72]。一旦对某种 NPC 语言  $L$  构造了一个 ZK 协议  $(P, V)$ , 显然对任意 NP 语言  $L'$ , 通过下面两步用 ZK 可以证明成员归属  $y \in L'$ :

1.  $P$  将  $y \in L'$  归约到  $x \in L$ , 其中  $L$  是一种 NPC 语言(例如,  $x$  是图的三色问题或布尔表达式的可满足性问题的一个实例)。由于  $P$  知道  $y \in L'$ , 该归约变换可以由  $P$  在关于  $y$  的长度的多项式时间内完成。 $P$  加密该变换并将密文发送给  $V$ 。
2.  $P$  执行一个 ZK 证明, 以便  $V$  验证正确加密了该多项式归约变换。如果加密采用的是 Goldwasser-Micali 概率加密方案(见算法 14.1), 关于正确加密了一个串的 ZK 证明可以很容易完成。在 18.2.2 节中我们将对此给出一个有说服力的解释。

不难看出, 这两步组合了证明成员归属  $x \in L$  的具体 ZK 协议, 确实构成了对  $y \in L'$  有效的 ZK 证明。注意除了要求  $L'$  属于  $NP$  以外, 这种方法对 NP 语言  $L'$  没有加以任何限制。

显然, 对任意 NP 语言的这种通用证明方法并不具有实用的效率。在 18.6 节中, 我们要求实际有效的 ZK(和 IP)协议的交互次数应当以一个安全参数的线性函数为界。通用的证明方法很难使其交互次数以线性多项式为界, 因为目前我们并不知道任何一种线性归约方法将 NP 问题转化为 NPC 问题。所有已知的归约都是很高次数的多项式。这就是为什么我们说对任意 NP 语言中成员归属的 ZK 证明只是一个理论结果, 虽然这是一个重要的结果, 为  $NP \subseteq PP$  提供了构造性证据。

等式  $NP = PP$  是计算复杂性理论中的公开问题。

### 18.3 零知识特性

现在考虑问题  $I$ (见 18.1 节)得到理想回答的情形:  $(P, V)$  是一个 ZK 协议, 也就是说除了  $P$  的断言是有效的以外, 关于  $P$  的秘密输入, 协议只向  $\bar{V}$ (或  $V$ ) 泄漏了零信息或没有泄漏任何信息。

为了使  $(P, V)$  具有这一性质, 必须限制  $V$ (和  $\bar{V}$ ) 的计算能力以公共输入长度的一个多项式为界。显然, 要是没有这样的限制, 就不必谈论零知识了, 因为具有无限计算资源的  $V$  自己能够找到隐藏在公共输入背后的秘密输入。

接下来的几小节将区分 ZK 的几个性质:

- 完备 ZK(18.3.1 节)
- 诚实验证者的 ZK(18.3.2 节)
- 计算 ZK(18.3.3 节)
- 统计 ZK(18.3.4 节)



### 18.3.1 完备零知识

设  $(P, V)$  是关于语言  $L$  的一个 IP 协议。对任意  $x \in L$ , 执行  $(P, V)(x)$  的证明不只是输出 Accept, 还会输出一个证明副本, 它交替插入示证者和验证者的副本。该证明副本中的元素是关于所有输入值的一个随机变量, 包括对  $(P, V)$  的随机输入。

显然, 如果  $(P, V)(x)$  泄漏了关于  $P$  的秘密输入的任何信息, 那么只可能是证明副本泄漏了信息。

但是, 如果证明副本中的随机变量在各自的概率空间中是均匀随机的, 并且独立于公共输入, 那么就不能说是它们导致了信息泄漏。在这种情况下 (即证明副本中的随机变量在各自的概率空间是均匀随机的, 并且独立于公共输入), 我们可以认为, 示证者向验证者讲话时用了一种没有包含冗余或具有最大可能熵 (见 3.7.1 节中熵的性质) 的语言。因此, 无论验证者多么聪明 (或强大), 即使用非常非常长的时间去学习这种语言, 也不可能知道这种语言所传达的任何内容。

现在证明协议 18.1 是完备 ZK 的。

**例 18.2** 回顾协议 18.1。运行证明  $(\text{Alice}, \text{Bob})(X)$  产生的证明副本为:

$$\text{Commit}_1, \text{Challenge}_1, \text{Response}_1, \dots, \text{Commit}_m, \text{Challenge}_m, \text{Response}_m$$

其中  $(i = 1, 2, \dots, m)$ 。

- $\text{Commit}_i = f(k_i)$ , 其中  $k_i \in_U \mathbb{Z}_n$ ;

显然, 由于 Alice 选取了等可能的  $k_i$ ,  $\text{Commit}_i$  在函数  $f$  的像空间内必然是等可能的, 且独立于公共输入  $X$ 。

- $\text{Challenge}_i \in \{0, 1\}$ ;

Bob 应当等可能地选取询问比特, 但我们不必要求他这样做, 看下面的应答;

- $\text{Response}_i = k_i + z \cdot \text{Challenge}_i \pmod{n}$ ;

显然, 由于  $k_i$  的等可能性, 对  $\text{Challenge}_i \in \{0, 1\}$  的任何情形 (即使  $\text{Challenge}_i$  是非均匀的),  $\text{Response}_i$  必然在  $\mathbb{Z}_n$  中均匀分布并独立于公共输入  $X$ 。

因此, 在协议 18.1 的运行过程中, Alice 发送的数据是均匀分布的, 它们不会告诉 Bob 关于 Alice 秘密输入的任何信息。该协议是完备 ZK 协议。  $\square$

从这个例子我们还看到, 无论 Bob 如何选取他的询问比特, Alice 的副本中的元素都是均匀分布的, 换句话说, Bob 没有办法影响 Alice 的副本的分布。因此, 即使 Bob 是不诚实的, 协议 18.1 仍然是完备 ZK 的。

对完备 ZK 协议, 不运行它我们也可以得到证明副本, 这样的副本 (它只是一个串) 可以通过随机抛掷硬币, 在关于副本长度的多项式时间内生成。定义 18.2 给出了完备 ZK 性这一重要概念。

**定义 18.2** 关于  $L$  的 IP 协议  $(P, V)$  称为是完备零知识的, 如果对任意  $x \in L$ ,  $(P, V)(x)$  的证明副本可以由一个 (关于输入长度的) 多项式时间算法  $\mathcal{EQ}(x)$  以相同的概率分布生成。

有效算法  $\mathcal{EQ}$  习惯上称为 ZK 协议的仿真器,它生成一个证明副本的仿真。但是,对  $(P, V)$  是完备 ZK 的情况,我们不想称  $\mathcal{EQ}$  为仿真器,它确切地是一个等同器。

### 18.3.1.1 Schnorr 身份识别协议

在协议 18.1 中, Bob 使用了比特询问,得到一个很高的正确性差错概率值  $\delta = 1/2$ 。因此,为了将差错概率缩小到  $2^{-m}$ ,协议必须重复  $m$  次。为了高度可信地对付 Alice 的欺骗,典型地要求  $m = 100$ 。这需要大量的交互,意味着在通信和计算上的性能都很差。

在一定条件下设置公共输入中的安全参数有可能降低差错概率值,从而减少交互的次数。该条件是验证者 Bob 必须知道  $n$  的分解。18.6.1 节将解释为什么需要这个条件。Bob 知道  $n$  的分解的一种特殊情形是  $n$  为素数。我们现在来看使用这种参数设置的一个具体协议,也就是由 Schnorr 提出的 **Schnorr 身份识别协议**[258],用于实际应用(基于智能卡)的身份识别。

Schnorr 身份识别协议是协议 18.1 的一种特殊情形,其中函数  $f(x)$  用有限域  $\mathbb{F}_p$  中的  $g^{-x} \pmod{p}$  实现,子群  $\langle g \rangle$  有素数阶  $q \mid p-1$ 。显然,  $g^{-x} \pmod{p}$  是同态的,而且对充分大的素数  $p$  和  $q$ ,例如  $|p| = 1024, |q| = 160$ ,根据 DL 假设(见 8.4 节中的假设 8.2),  $g^{-x} \pmod{p}$  也是单向的。

在这种参数环境下,协议 18.2 描述的 Schnorr 身份识别协议允许 Bob 使用扩展到  $\log_2 \log_2 p$  比特的询问。

#### 协议 18.2 Schnorr 身份识别协议

公共输入:

$p, q$ : 两个素数,满足  $q \mid p-1$ ;

(\* 典型的长度设置:  $|p| = 1024, |q| = 160$  \*)

$g$ :  $\text{ord}_p(g) = q$ ;

$y$ :  $y = g^{-a} \pmod{p}$ ;

(\* 数组  $(p, q, g, y)$  是 Alice 的公钥材料,由某个 CA 发放证书 \*)

Alice 的秘密输入:  $a < q$ ;

对 Bob 的输出: Alice 知道某个  $a \in \mathbb{Z}_q$ , 满足  $y \equiv g^{-a} \pmod{p}$ 。

重复下列步骤  $\log_2 \log_2 p$  次:

1. Alice 选取  $k \in \mathbb{Z}_q$ , 计算  $\text{Commit} \leftarrow g^k \pmod{p}$ , 发送 Commit 给 Bob;
2. Bob 选取  $\text{Challenge} \in_U \{0, 1\}^{\log_2 \log_2 p}$ , 发送 Challenge 给 Alice;
3. Alice 计算  $\text{Response} \leftarrow k + a \cdot \text{Challenge} \pmod{q}$ , 发送 Response 给 Bob;
4. Bob 检验  $\text{Commit} \equiv g^{\text{Response}} y^{\text{Challenge}} \pmod{p}$ ;

如果验证显示错误则拒绝并中止协议;

Bob 接受证明。

(\* Bob 应该应用算法 15.2 计算  $g^{\text{Response}} y^{\text{Challenge}} \pmod{p}$ , 这样开销近似于单底数模指数运算 \*)

**注释 18.2** 如果公开素数  $q \mid p-1$ , Schnorr 身份识别协议就不再是一个回答子群成员归属问题的协议了。此时通过检验  $y^q \equiv g^q \equiv 1 \pmod{p}$ , 无需 Alice 的帮助 Bob 自己就能够独自

回答问题  $y \in \langle g \rangle$  与否。因此, Schnorr 身份识别协议是在证明一个更具体的问题: Alice 拥有以  $g$  为底  $y$  的离散对数作为她的密码证书。  $\square$

现在考察 Schnorr 身份识别协议的安全性。

### 18.3.1.2 Schnorr 身份识别协议的安全性

#### 完全性

证明简单, 在这里就不给出证明。事实上可以得到  $\epsilon = 1$ 。留给读者作为一个习题(见习题 18.7)。

#### 正确性

假设 Alice 是一个欺骗者, 即她没有正确的离散对数。对 Alice 在交互中发送的 Commit, Bob 选取  $\text{Challenge} \in_U \{0, 1\}^{\log_2 \log_2 p}$  后, 等待

$$\text{Response} = \log_g [\text{Commit} y^{\text{Challenge}} (\bmod p)] (\bmod q)$$

这一等式表明, 对固定的 Commit 和  $y$ , 有  $\log_2 p$  个不同应答值分别与  $\log_2 p$  个不同询问值相对应。给定数量不大的  $\log_2 p$ , 从  $\text{Commit} y^{\text{Challenge}} (\bmod p)$  计算正确应答的最好策略是在固定承诺 Commit 之前如下猜测询问 Challenge:

1. 选取  $\text{Response} \in_U \mathbb{Z}_q$ ;
2. 猜测  $\text{Challenge} \in_U \{0, 1\}^{\log_2 \log_2 p}$ ;
3. 计算  $\text{Commit} \leftarrow g^{\text{Response}} y^{\text{Challenge}} (\bmod p)$ 。

显然, 对每一次交互, 猜测正确的正确性概率为  $1/\log_2 p$ , 也就是说, 我们已经求得单轮消息交互的正确性差错概率是  $\delta = 1/\log_2 p$ 。

在 Schnorr 身份识别协议中, 单轮消息交互的正确性差错概率降低意味着协议 18.1 的性能提高。这是因为, 对协议 18.1 运行  $m$  次得到可忽略的正确性差错概率  $\delta = 2^{-m}$ , Schnorr 身份识别协议只需要

$$\ell = \frac{m}{\log_2 \log_2 p}$$

轮交互, 而协议 18.1 进行  $m$  轮交互的正确性差错概率保持不变。

对  $p \approx 2^{1024}$  和  $m = 100$ , 我们有  $\ell = 100/10 = 10$ 。也就是说, 扩展提问减少了询问次数, 为协议 18.1 交互次数的  $1/10$ , 同时保持了同样低的差错概率。

#### 完备 ZK 性

对公共输入  $y$ , 可以构造一个(关于  $|p|$  的)多项式时间等同器  $\mathcal{EQ}(y)$  如下:

1.  $\mathcal{EQ}$  初始化 Transcript 为一个空串;
2. 对  $i = 1, 2, \dots, \log_2 \log_2 p$ ;
  - a)  $\mathcal{EQ}$  选取  $\text{Response}_i \in_U \langle g \rangle$ ;
  - b)  $\mathcal{EQ}$  选取  $\text{Challenge}_i \in_U \{0, 1\}^{\log_2 \log_2 p}$ ;
  - c)  $\mathcal{EQ}$  计算  $\text{Commit}_i \leftarrow g^{\text{Response}_i} y^{\text{Challenge}_i} (\bmod p)$ ;
  - d)  $\text{Transcript} \leftarrow \text{Transcript} \parallel \text{Commit}_i, \text{Challenge}_i, \text{Response}_i$ 。

显然, Transcript 可以在多项式时间内生成, 并且其中的元素和实际证明副本中的元素具有同样的分布。

根据对 Schnorr 身份识别协议的分析, 我们看到增加询问的长度减少了交互的次数, 而正确性差错概率保持不变。那么为什么要限制扩展的长度为一个相当古怪而又不大的值  $\log_2 \log_2 p$  呢?

增加询问长度不只是提高性能(正面的结果), 在 18.3.2 节中将进一步看到它也有负面的结果。小心提问长度问题!

### 18.3.2 诚实验证者的零知识

乍一看 Schnorr 身份识别协议, 弄不明白为什么要把询问比特的长度限制为  $|\text{Challenge}| = \log_2 \log_2 p$ 。似乎如果我们使用  $|\text{Challenge}| = \log_2 p$ , 协议效率会更高: 为了获得同样低的正确性概率 ( $\delta \approx 1/p$ ) 防止 Alice 欺骗, 只需要一轮交互。而且, 似乎对 Schnorr 身份识别协议的等同器  $\mathcal{EQ}$  也可以用同样的方式构造, 此时  $\mathcal{EQ}$  生成包含均匀分布元素的副本也只需要一次“循环”。

不过, 这个问题有点微妙, 现在我们就来考察它。

#### 18.3.2.1 不诚实的验证者所能做的

假设 Bob 是一个不诚实的验证者, 也就是说, 他会不遵循协议指令, 总是试图蒙骗 Alice 泄漏一些对他有用的信息。假定允许 Bob 选取一个很大的 Challenge, 使得  $2^{\text{Challenge}}$  是一个非多项式有界量。那么他就可以设计一个圈套, 迫使 Alice 生成一个在多项式时间内不可等同(不能等同)或不可仿真的副本。如果 Bob 能够做到这一点, 那么根据定义 18.2, 协议不再是完备 ZK 的。

现在来检验这个问题。稍微修改一下 Schnorr 身份识别协议, 允许 Bob 选择  $\text{Challenge} \in \mathbb{Z}_q$ , 即将询问空间由  $\{0, 1\}^{\log_2 \log_2 p}$  扩大到  $\mathbb{Z}_q$ 。下面是 Bob 在这个修改后的 Schnorr 身份识别协议中需要做的。

一旦接收到 Commit, Bob 运用具有很大大输出空间  $\mathbb{Z}_q$  的适当伪随机函数 prf 如下生成 Challenge:

$$\text{Challenge} \leftarrow \text{prf}(\text{"有意义的副本, Alice 签署"} \parallel \text{Commit})$$

这样生成的 Challenge 是伪随机的(即不是真随机的)。稍后就会明白字符串“有意义的副本, Alice 签署”的全部用意。

由于伪随机性和真随机性之间一般不可区分, 可怜的 Alice 无法认识到 Challenge 的伪随机性, 因此不得不遵循协议指令返回  $\text{Response} = k + a \text{ Challenge} \pmod{q}$ 。

记住 Alice 的应答满足

$$\text{Commit} = g^{\text{Response}} y^{\text{Challenge}} \pmod{p} \quad (18.3.1)$$

这正是 Bob 执行的验证程序, 所以 Alice 帮助 Bob 构造了下列等式:

$$\text{Challenge} = \text{prf}(\text{"有意义的副本, Alice 签署"} \parallel g^{\text{Response}} y^{\text{Challenge}} \pmod{p}) \quad (18.3.2)$$

在第三方看来, 式(18.3.2)意味着下面两种情形之一:

- i) 该等式由 Alice 用她的秘密输入构造。因此 Alice 展示了一个事实: 她和 Bob 交互了并受到了 Bob 愚弄, 或者
- ii) Bob 已经成功破解了具有很大大输出空间  $\mathbb{Z}_q$  的伪随机函数 prf, 因为他已经构造了等式

$$\text{Challenge} = \text{prf}(\dots y^{\text{Challenge}})$$

由于 prf 被认为是单向的,所以这是一个著名的困难问题。

假定 Bob 是多项式有界的,第三方当然相信是情形(i)。Alice 够可怜的,满足式(18.3.1)和式(18.3.2)的证明副本(Commit, Challenge, Response)中,二元对(Commit, Response)正好是 Schnorr 签名方案下对消息“有意义的副本, Alice 签署”的一个签名(用  $\text{prf} = H$  检验算法 10.4)! 既然只有 Alice 才能发布该签名(回忆一下,在 16.3.2 节我们已经证明了该签名方案在适应性选择消息攻击下防伪造的强安全性),第三方做出了正确判断!

使 Alice 略感欣慰的是,由 Bob 造成的信息暴露还不是太严重(尽管这一说法必须基于实际应用)。正如我们在 7.5.2 节中已经分析的,如果 Alice 选取独立于以前所有实例的  $k \in_U \mathbb{Z}_q$ , 那么

$$\text{Response} = k + a \text{ Challenge} \pmod{q}$$

构成 Alice 秘密输入  $a$  的一次一密式(移位密码)加密,它提供信息论安全性。这意味着该证明副本仍然没有向 Bob 或第三方泄漏关于 Alice 秘密输入的任何信息。

不过,随着交互式证明退化为不必以交互方式发布的签名,也就失去了由交互式证明所提供的安全服务:现在任何第三方都能够验证证明结果。这意味着此时的知识证明不再是“在黑暗中”进行的,而是“公开”进行的。这就是为什么变形协议(即采用大规模询问的 Schnorr 身份识别协议)已经不再是 ZK 的原因!

一般来说,如果 Schnorr 身份识别协议使用  $\mathbb{Z}_q$  中的大规模询问,那么协议具有验证者诚实的零知识性质。在验证者诚实的 ZK 协议中,如果验证者诚实地遵循协议指令,那么协议是完备 ZK 的。这是因为如果验证者选取真随机询问,那么证明副本可以有效地等同。

对于验证者诚实的 ZK 协议  $(P, V)$ ,如果限制  $V$  的行为方式,使得它不能迫使  $P$  生成不可等同的或不可仿真的副本,那么  $(P, \tilde{V})$  仍然可以是一个完备 ZK 协议。在 18.3.2.3 节我们将看到限制询问比特的多少是一种解决方法。还有多种方法对  $V$  的行为施加限制,例如

- 一种解决办法是迫使  $V$  证明它在选择随机询问中是诚实的;在 18.6.2 节我们将介绍一个效率极高的 ZK 证明,它采用了这种思想;
- 向  $V$  提供仿真“证明”的资格,因此,如果不诚实的验证者试图欺骗证明者,只能表明它是不诚实的;在 18.7.1 节我们将另外介绍一个效率极高的 ZK 证明,它采用的就是这种想法。

### 18.3.2.2 Fiat-Shamir 启发式

Fiat 和 Shamir 建议了一种一般方法,将验证者诚实的安全 ZK 协议转化为数字签名方案 [110]。该方法使用的正是不诚实的验证者的攻击技术,我们在 18.3.2.1 节中已经了解了这种技术。通常用(Commit, Challenge, Response)表示一个验证者诚实的 ZK 协议的副本,为了构造消息  $M \in \{0,1\}^*$  的数字签名,该转化方法要利用一个适当的杂凑函数  $H$ :

$$\text{Challenge} \leftarrow H(M \parallel \text{commit})$$

这种一般方法称为 **Fiat-Shamir 启发式**。

很容易明白,三元组的 ElGamal 族签名方案(见 16.3.1 节)是由 Fiat-Shamir 启发式生成的签名方案的特殊情况。事实上,对三元组的 ElGamal 族签名方案强不可伪造性的形式化安全

性证明技术适用于采用 Fiat-Shamir 启发式,由验证者诚实的 ZK 协议转化来的所有签名方案。

Fiat-Shamir 启发式显然是可以公开验证的,由于这一事实,可以像验证数字签名一样验证隐藏在单向函数中的断言(如成员归属或论据隐藏断言),即它不是“在黑暗中的证明”。这种方式证明断言通常称为知识证明。由于我们在 16.3.2 节中建立的强安全性结论(适应性选择消息下的不可伪造性),知识证明仍然是证明隐藏在单向函数中的断言很好也很有用的方式。

在某些应用中,如证明一个秘密具有要求的结构,“在黑暗中证明”并不是一个根本的安全要求(即证明者并不觉得需要拒绝参与交互)。知识证明在这类应用中是一个非常有用并且恰当的概念。

### 18.3.2.3 回到完备零知识

现在考虑和不诚实的验证者运行 Schnorr 身份识别协议(注意,不是使用大规模询问比特的变型)的情形,其中验证者试图蒙骗 Alice 发布一个 Schnorr 签名方案下的签名。

而现在对于任何输出为  $\log_2 \log_2 p$  比特的伪随机函数 prf,任何人都可以有效地构造等式(18.3.2),也就是说,证明副本可以有效地等同。我们来看如何等同,其效率如何。

设  $\mathcal{EQ}$  是一个等同器。 $\mathcal{EQ}$  要做的就是随机选取  $\text{Response} \in {}_U \mathbb{Z}_q$ , 对固定的  $\text{Challenge} \in \{0,1\}^{\log_2 \log_2 p}$ , 验证式(18.3.2)是否成立。如果验证失败,  $\mathcal{EQ}$  简单地尝试另外一个  $\text{Response} \in {}_U \mathbb{Z}_q$ 。在穷举完 prf 的  $\log_2 \log_2 p$  比特输出空间之前,尝试-错误检验将会成功。由于 prf 的长度只有  $\log_2 \log_2 p$ , 其输出空间可以在  $\log_2 p$  步内穷举完,也就是说,在关于  $p$  长度的(线性)多项式时间内。

一旦等式成立,  $\mathcal{EQ}$  可以用式(18.3.1)设置 Commit, 因此

$$\text{Transcript} = \text{Commit}, \text{Challenge}, \text{Response}$$

是模仿单轮交互的一个等同“证明副本”,并在关于  $p$  长度(即关于  $\log_2 p$ )的多项式时间内生成。该等同的“证明副本”满足

$$\text{Challenge} = \text{prf}(\text{“有意义的副本, Alice 签署”} \parallel \text{Commit})$$

和

$$\text{Commit} \equiv g^{\text{Response}} y^{\text{Challenge}} \pmod{p}$$

不过它根本不是一个有意义的副本,我们也已经看到它不一定是 Alice 发布的。

到目前为止,我们已经知道,对于 ZK 协议中的提问比特,其长度确实是个问题。

### 18.3.3 计算零知识

我们已经看到,为了证明一个 IP 协议  $(P, \tilde{V})$  是完备 ZK 的,我们必须构造一个等同器:它能够有效地生成“证明”副本,其概率分布和  $(P, \tilde{V})$  生成的相同。对于计算零知识的 IP 协议,可以放宽这一要求。

**定义 18.3** 关于  $L$  的一个 IP 协议  $(P, V)$  称为是计算 ZK 的,如果对任意  $x \in L$ ,  $(P, V)(x)$  的证明副本可以由一个(关于输入长度的)多项式时间算法  $S(x)$  仿真,其概率分布与证明副本是多项式不可区分的。

关于该定义中多项式不可区分的概念在定义 4.15 中。



为了理解计算 ZK 协议,我们用另外一种方式修改协议 18.1。在这一修改中,同态单向函数定义在一个未知大小的空间上,也就是说,现在  $\mathbb{Z}_n$  中的  $n$  对  $P$  和  $V$  来说都是一个秘密整数。在一个秘密定义域上构造  $f$  是可能的,这里就有一个具体构造。

### 18.3.3.1 构造同态单向函数 $f(x)$

设  $P$  和  $V$  协商了一个非常大的随机奇合数  $N$ ,无人知道  $N$  的分解。如果在  $N$  的协商中双方都随机输入,这就很容易,在这里我们省略这一细节。他们也可以类似地协商一个随机元素  $a < N$ ,使得  $\gcd(a, N) = 1$ 。

由于  $N$  很大并且是随机的,  $N$  以压倒性的概率含有  $P$  和  $V$  都不知道的大素因子  $p$ , 进一步,  $p-1$  也应当含有  $P$  和  $V$  都不知道的大素因子  $q$ 。我们不去研究这一压倒性的概率是多少,但要提醒读者,存在这样的大素数  $p$  和  $q$  正是随机大合数分解困难的原因(读者回顾 8.8 节就会有所启发)。

同样,由于  $N$  和  $a$  是随机协商的,所以乘法阶  $\text{ord}_N(a)$  也以压倒性的概率是一个大的秘密整数。我们相信这是“压倒性的”:因为对任意  $q \mid \phi(N)$ , 在  $\mathbb{Z}_n^*$  中至多有  $1/q$  的元素的阶和  $q$  互素,所以  $q \mid \text{ord}_N(a)$  的概率至少是  $1 - 1/q$ 。

现在对任意  $x \in \mathbb{Z}_N$ ,  $P$  和  $V$  “定义”

$$f(x) \stackrel{\text{def}}{=} a^x \pmod{N} \quad (18.3.3)$$

注意到我们这里用的加引号的“定义”,因为这个函数的定义域不可能是  $\mathbb{Z}_N$ , 而是  $\mathbb{Z}_{\text{ord}_N(a)}$ , 即对任意  $x \in \mathbb{Z}_N$ ,

$$f(x) = f(x \pmod{\text{ord}_N(a)})$$

恒成立,换句话说,  $f$  的输入总是来自于小于  $\mathbb{Z}_N$  的空间  $\mathbb{Z}_{\text{ord}_N(a)}$ 。

不难看出  $f(x)$  仍然是同态单向的。同态性一般视为

$$f(x+y) = a^{x+y} = a^x \cdot a^y \pmod{N}$$

单向性基于模  $p$  的离散对数问题(回忆一下,未知大素数  $p \mid N$ ): 由  $f(x) = f(1)^x \pmod{N}$  求  $x$  必定比由  $f(1)^x \pmod{p}$  求  $x \pmod{p-1}$  困难,而根据离散对数假设(假设 8.2),  $f(1)^x \pmod{p}$  是单向的。

### 18.3.3.2 计算零知识协议

运用 18.3.3.1 节中构造的  $f(x)$ , 我们可以构造一个计算 ZK 协议。

**例 18.3** 在协议 18.1 中使用 18.3.3.1 节中构造的单向同态函数  $f(x)$ , 即  $f(x)$  定义在式 (18.3.3) 中, 设 (Alice, Bob) 就是协议 18.1 的这样一个变形。

既然 Alice 不再知道  $n = \text{ord}_N(a)$ , 她就再也不能均匀分布地取样  $\mathbb{Z}_{\text{ord}_N(a)}$  中的随机数。为了使 Alice 仍然能够进行证明(即保持完全性), 对 Alice 的协议指令需要稍加修改, 例如以如下的方式(设  $z < N$  是 Alice 的秘密输入)修改

1. Alice 选取  $k \in_U \mathbb{Z}_{N^2-z}$ , 计算并发送  $\text{commit} \leftarrow f(k)$  给 Bob;

2. Bob...(\* 不变 \*);
3. Alice 计算并发送  $\text{Response} \leftarrow \begin{cases} k & \text{如果 Challenge} = 0 \\ k + z & \text{如果 Challenge} = 1 \end{cases}$  给 Bob;
4. Bob...(\* 不变 \*).

在这一修改中,对 Bob 的指令没变,但对 Alice 的指令有两处变化。在第 1 步,随机值  $k$  从  $\mathbb{Z}_{N^2-z}$  中取样。稍后我们会解释为什么她必须从这个相当特殊的空间选取  $k$ 。在第 3 步(对 Challenge = 1 的情形),她采用整数空间  $\mathbb{Z}$  中的加法计算  $\text{Response}(\leftarrow k + z)$ ,即不进行模约简。现在由于她没有该运算需要的模数  $n = \text{ord}_N(a)$ ,也就不能进行模约简运算了。

该变形的完全性和正确性可以像例 18.1 那样类似地推导。

不过,因为现在不能构造一个有效的等同器,使得它生成的“证明”副本和  $(\text{Alice}, \tilde{\text{Bob}})(X)$  生成的副本具有同样的分布,所以再也不能证明该变型是完备 ZK 的。

事实上,通常的仿真器将生成一个分布不同的副本。仿真器  $S$  在这类仿真中执行下列步骤:

1.  $S$  选取  $\text{Response} \in_U \mathbb{Z}_{N^2}$ ;
2.  $S$  选取  $\text{Challenge} \in_U \{0, 1\}$ ;
3.  $S$  计算  $\text{Commit} \leftarrow f(\text{Response}) / X^{\text{challenge}} \pmod{N}$ 。

显然,(对 Challenge = 1 的情形)尽管证明中的 Response 在区间  $[z, N^2)$  中均匀分布,仿真副本却是在区间  $[0, N^2)$  中均匀分布,它们的分布是不同的。没有  $z$ ,  $S$  不可能等同 Alice 的行为。

然而变形  $(\text{Alice}, \tilde{\text{Bob}})$  是计算 ZK 的。这是因为,对  $z < N$ ,这两个不同的分布  $x \in_U [z, N^2)$  和  $y \in_U [0, N^2)$  在计算上是不可区分的。由

$$\text{Prob}[y \leq z < N \mid y \in_U [0, N^2)] < \frac{N}{N^2} = \frac{1}{N} \quad (18.3.4)$$

我们有

$$|\text{Prob}[\text{Response} \in_U [z, N^2)] - \text{Prob}[\text{Response} \in_U [0, N^2)]| < \frac{1}{N}$$

根据定义 4.15(见 4.7 节),证明副本和仿真副本中的 Response 在计算上是不可区分的。因此,我们已经构造了一个多项式时间仿真器  $S$ ,或者说,根据定义 18.3,  $(\text{Alice}, \tilde{\text{Bob}})$  是计算 ZK 的。□

现在我们可以解释为什么 Alice 必须从相当特殊的空间中  $\mathbb{Z}_{N^2-z}$  选取承诺  $k$ 。

首先,  $N^2 - z$  中的  $-z$  部分是必要的,否则由于加法没有模约简,最后 Response 可能会超过  $N^2$ 。如果出现这种情况,该协议在任何意义上都不能称为是 ZK 的。

其次,  $N^2 - z$  中的  $N^2$  部分是为了得到概率界 (18.3.4),这样该协议才能够获得计算 ZK 性。事实上,  $N^2$  已是不必要地过大了,对任意常数  $\alpha > 0$ ,计算上的 ZK 可以由  $N^{1+\alpha}$  得到。鼓励读者予以证实(提示:观察式 (18.3.4) 的右边,用  $\frac{1}{N}$  代替  $\frac{1}{N^2}$ )。

在 ZK 协议(例如 Schnorr 身份识别协议)的实际应用中,绝大多数单向函数用可获得的公钥密码技术实现(例如 18.3.3.1 节或 Schnorr 身份识别协议中的  $f(x)$ )。因此,计算 ZK 是 ZK (和 IP)协议中最重要且充分(即适合应用的)的概念。

### 18.3.4 统计零知识

Goldwasser、Micali 和 Rackoff [128] 还引入了统计零知识概念。一个 IP 协议是统计 ZK 的,

如果存在一个有效的仿真器仿真证明副本,并达到用任何统计分辨器都不能区分的精度。统计分辨器类似于定义 4.14 中定义的多项式分辨器,不过它的运行时间不必是多项式有界的。从这个差别我们可以看出,与计算 ZK 协议相比,统计 ZK 协议具有更严格的 ZK 性质。

事实上,例 18.3 中的计算 ZK 协议(Alice, Bob)是统计 ZK 的。这是因为,式(18.3.4)说明下述事件发生的概率小于一个可忽略量  $1/N$ :

仿真副本中的 Response 小于  $z$

因此,Response 在两个副本中都大于  $z$  并且都均匀分布的概率至少是  $(N-1)/N$ 。任何分辨器即使永远运行下去也不能区分它们!

统计 ZK 和计算 ZK 在概念上没有本质区别。但是,由于前者是一个更严格的安全概念,所以如果协议设计者能够做到的话,在实际应用中更希望构造的协议是统计 ZK 的。

## 18.4 证明还是论据

我们已经明确推断了一个 IP 协议( $P, V$ ),为了具有(到目前为止介绍的 4 个 ZK 概念中任何一个的)ZK 性质, $V$  和  $\tilde{V}$  的计算能力必须以关于公共输入长度的一个多项式为界。然而,到目前为止我们都一直不很明确关于  $P$  或  $\tilde{P}$  的计算能力。

### 18.4.1 零知识论据

细心的读者可能已经注意到,对于到目前为止介绍的所有 ZK 协议,实际上要求  $P$  和  $\tilde{P}$  具有多项式有界的计算能力。事实上,在推导这些协议的正确性时,总是一开始就说“如果  $P$ (或  $\tilde{P}$ )不知道  $X$  的原像……”

对于  $IP = PP$  中的语言,这种“如果……”实际上蕴涵  $P$ (或  $\tilde{P}$ )是多项式有界的。如果我们说  $P$  是无界的,它能够求得单向函数  $f$  的原像,那么所有这些协议的正确性推导都是无效的。显然,对任意 Challenge,无界的  $P$  和  $\tilde{P}$  可以这样求得 Response:

$$\text{Response} \leftarrow \text{pre-image}(\text{Commit}, X^{\text{challenge}})$$

如果计算能力无界的算法用这种方式求得原像,我们永远不能估计式(18.2.3)的正确性概率  $\delta$ 。到目前为止所介绍的协议进行正确性推导的所有情形中, $\delta$  值都是在  $P$ (和  $\tilde{P}$ )有界的(不明确)假定下得到的。

如果语言  $L$  的一个 ZK 协议( $P, V$ )要求  $P$ (和  $\tilde{P}$ )具有(关于输入长度的)多项式有界的计算能力,那么( $P, V$ )称为一个零知识论据协议。为了确立协议的正确性,通常需要这一要求。论据没有证明严格,尤其是当  $P$  是一个无界实体时,它没有什么意义。

现在我们已经清楚完备的、验证者诚实的、计算的以及统计的 ZK 论据协议了。Schnorr 身份识别协议也是一个 ZK 论据协议。实际上,我们还没有碰到任何零知识证明协议。

在继续描述 ZK 证明协议之前,我们要澄清很重要的一点。在绝大多数实际应用中,也就是通常情况下运用基于复杂性理论的现代密码技术保护信息的情形,安全系统中主体(包括 ZK 协议的示证者)的计算资源极可能是多项式有界的,它们不可能很快解答 NP 问题。所以, ZK 论据仍然是一个非常有用的概念。

## 18.4.2 零知识证明

在零知识证明协议中,确立正确性无须要求  $P$  或  $\tilde{P}$  是多项式有界的。

现在来看一个 ZK 证明协议。二次剩余的证明提供了一个很好的 ZK 证明协议的例子,它也是一个成员归属问题:对奇合数  $N, x \in \text{QR}_N$ 。

### 18.4.2.1 二次剩余的 ZK 证明

设  $N$  是一个大的奇合数,至少有两个不同的素因子。我们在 6.5 节已经学习了模整数的二次剩余,并知道以下数论事实:

**事实 1** 已知  $N$  的分解,对任意  $x \in \text{QR}_N$ ,可以有效求得  $x$  模  $N$  的平方根  $y$ ,满足  $y^2 \equiv x \pmod{N}$ 。这可以用算法 6.5。

**事实 2** 对任意  $x \in \text{QNR}_N$  (非二次剩余),  $\mathbb{Z}_N^*$  中不存在  $x$  的平方根(算法 6.5 的第 1 步无法运行)。

**事实 3** 如果  $x \in \text{QNR}_N$ ,那么  $x \cdot y \in \text{QR}_N$  意味着  $y \in \text{QNR}_N$  (通过检验  $x, y$  和  $x \cdot y$  的 Jacobi 符号的所有情况,读者可以证实这一点)。

运用这些事实可以构造一个完备 ZK 证明协议,让 Alice 向 Bob 证明一个数是模一个奇合数的二次剩余。这一协议由 Goldwasser、Micali 和 Rackoff [128] 提出,在协议 18.3 中具体描述。

让我们首先分析协议 18.3 的正确性。

### 协议 18.3 二次剩余的完备零知识证明协议

公共输入:

$N$ : 一个大的奇合数,不是一个素数的幂;

$x$ :  $\text{QR}_N$  中的一个元素。

Alice 的秘密输入:

$y \in \mathbb{Z}_N^* : y^2 \equiv x \pmod{N}$ ;

向 Bob 输出:  $x \in \text{QR}_N$ 。

重复下列步骤  $m$  次:

1. Alice 选取  $u \in {}_U \text{QR}_N$ , 计算  $\text{Commit} \leftarrow u^2 \pmod{N}$ , 发送 Commit 给 Bob;

2. Bob 选取  $\text{Challenge} \in {}_U \{0, 1\}$ ; 发送 Challenge 给 Alice;

3. Alice 计算  $\text{Response} \leftarrow \begin{cases} u & \text{如果 Challenge} = 0 \\ u y \pmod{N} & \text{如果 Challenge} = 1 \end{cases}$ , 发送 Response 给 Bob;

4. Bob 验证:

$$\text{Response}^2 \pmod{N} \equiv \begin{cases} \text{Commit} & \text{如果 Challenge} = 0 \\ \text{Commit} x \pmod{N} & \text{如果 Challenge} = 1 \end{cases}$$

我如果验证失败, Bob 拒绝并中止协议;

Bob 接受证明。

### 正确性

假设  $x \in \text{QNR}_N$  (即和欺骗者 Alice 运行协议), 我们来求差错概率  $\delta$ 。当然, 现在我们假设 Alice 在计算上是无界的。

对 Challenge = 0, Bob 知道 Response 是 Commit 的平方根, 所以  $\text{Commit} \in \text{QR}_N$ 。

对 Challenge = 1, Bob 知道 Response 是 Commit  $x$  的平方根, 所以  $\text{Commit } x \in \text{QR}_N$ 。根据事实 3, Bob 进一步知道  $\text{Commit} \in \text{QNR}_N$ 。

因此, 如果  $x \in \text{QNR}_N$ , 那么 Bob 知道  $\text{Commit} \in \text{QR}_N$  或者  $\text{Commit} \in \text{QNR}_N$ , 分别依赖于他的随机询问比特 0 或 1。由于 Alice 在 Bob 选取随机询问比特之前发送 Commit, 所以 Alice 必然已经正确猜测到了 Bob 的询问比特。显然, 正确性差错概率  $\delta = 1/2$ 。因此, 通过 Bob 的验证  $m$  次后得到正确性概率为  $2^{-m}$ 。

由于事实 2, 即使 Alice 在计算上是无界的, 她也不能计算  $x \in \text{QNR}_N$  的平方根, 因此不得不猜测 Bob 的随机询问比特。所以, 对计算上无界的 Alice, 该正确性仍然成立。

### 完全性和完备零知识性

完全性从事实 1 直接得到。

完备零知识性可以通过构造一个等同器  $\mathcal{EQ}$  证明,  $\mathcal{EQ}$  生成等同的证明副本如下:

对  $i = 1, 2, \dots, m$

1.  $\mathcal{EQ}$  选取  $\text{Response}_i \in {}_U\mathbb{Z}_N^*$ ;
2.  $\mathcal{EQ}$  选取  $\text{Challenge}_i \in {}_U\{0, 1\}$ ;
3.  $\mathcal{EQ}$  设置  $\text{Commit}_i \leftarrow \begin{cases} \text{Response}_i^2 \pmod{N} & \text{如果 Challenge} = 0 \\ \text{Response}_i^2 / x \pmod{N} & \text{如果 Challenge} = 1 \end{cases}$

容易验证这个等同副本中的元素和证明副本中的元素分布相同。

#### 18.4.2.2 二次非剩余的 ZK 证明

ZK 证明非二次剩余的协议也可以用协议 18.3 中的思想构造。基本思路如下。

对公共输入  $x \in \text{QNR}_N$ , Bob 可以随机使用  $\text{Challenge} \leftarrow r^2 \pmod{N}$  或  $\text{Challenge}' \leftarrow xr^2 \pmod{N}$  向 Alice 询问, 其中  $r$  是  $\mathbb{Z}_N^*$  中的随机元素。显然,  $\text{Challenge} \in \text{QR}_N$ , Alice 能够确定它并回答 YES。另一方面, 如果实际上  $x$  属于  $\text{QNR}_N$ , 那么根据事实 3,  $\text{Challenge}' \in \text{QNR}_N$ ; Alice 也能确定它并回答 NO。

这样构造的随机元素要么属于  $\text{QR}_N$ , 要么属于  $\text{QNR}_N$ , Bob 用它们重复向 Alice 提问。根据 Alice 对这些随机提问一致的正确应答, Bob 可以验证  $x \in \text{QNR}_N$ 。这个协议的详细描述可以在 [128] 中找到。

ZK 证明二次剩余和非二次剩余在证明任意比特串加密 (使用 Goldwasser-Micali 概率加密算法, 算法 14.1) 的正确性方面有很好的应用, 该应用对于推导在 18.2.3 节中讨论的理论结果很有用。

## 18.5 双边差错协议

对于到目前为止学习的所有 ZK (证明或论据) 协议, 我们已经看到, 它们的完全性概率表

达式(18.2.2)的特征都是  $\epsilon = 1$ , 而正确性概率表达式(18.2.3)的特征都是  $\delta > 0$ 。由于  $\epsilon = 1$ , 这些协议具有完备的完全性, 也就是说, 如果示证者没有欺骗, 那么验证者总是接受证明。使用在 4.4 节中学习的表示随机化算法差错概率特征的术语, 可以称所有这些协议具有 Monte Carlo 子类(即“总是很快且可能正确的”子类, 见 4.4.3 节)中的单边差错。对这类协议, 单边差错可能发生在示证者(Alice)一侧, 也就是说, 尽管事实上  $x \notin L$ , Alice 还是可能欺骗并试图“证明”  $x \in L$ , Bob 也可能受到蒙骗, 从而接受她的“证明”(虽然通过一系列的独立重复证明可以做到正确性差错概率  $\delta$  任意小)。

有些 ZK 协议可能还有验证者(Bob)一侧的差错, 也就是说, 完全性差错概率表达式(18.2.2)的特征是  $\epsilon < 1$ , 称这类协议含有双边差错, 或者说属于 Atlantic City 子类(即属于“可能很快且可能正确的”子类, 见 4.4.5 节)。现在来看一个这种协议。

### 18.5.1 零知识证明双素整数

ZK 证明二次剩余的一个非常有用的应用是证明奇合数  $N$  恰有两个素因子, 即  $N \in E_{2\_Prime}$  或者说是有效的 RSA 模数。

在 4.7 节中, 语言  $E_{2\_Prime}$  称为一个系综, 这种语言中的任何元素都是一个奇合数, 并且有两个不同的素因子。也是在 4.7 节中, 我们认为这种语言和另外一个(种)有三个不同素因子的奇合数集合的系综(语言)  $E_{2\_Prime}$  是不可区分的。

假设 Alice 构造了一种语言  $N \in E_{2\_Prime}$  使得她知道分解(例如, 通过将两个不同素数相乘来构造), 她能够用完备 ZK 向 Bob 证明  $N \in E_{2\_Prime}$ 。这个证明要用到协议 18.3 中的三个数论事实, 以及下面另外两个事实:

**事实 4** 如果  $N \in E_{2\_Prime}$ , 那么

$$J_N(1) = \{x \mid x \in \mathbb{Z}_N^*, \left(\frac{x}{N}\right) = 1\}$$

中的元素恰好有一半是二次剩余, 即  $\#QR_N = \frac{\#J_N(1)}{2}$ 。这是因为这些元素模两个素因子只有一半有正的 Legendre 符号; 要使 Jacobi 符号为正, 另外一半模两个素因子的 Legendre 符号必定为负。

**事实 5** 如果  $N \notin E_{2\_Prime}$  且  $N$  不是素数或素数的幂, 那么  $J_N(1)$  至多有四分之一的元素为二次剩余, 即  $\#QR_N \leq \frac{\#J_N(1)}{2}$ 。这将事实 4 推广到了  $N$  有三个或更多个不同素因子的情形。注意, 要保证  $x$  属于  $QR_N$  的成员归属, 要求对所有  $p \mid N$ , 都有  $x(\bmod p) \in QR_p$ 。

在事实 5 中, 要求  $N$  不是素数的幂。如果  $N$  是一个素数的幂, 即  $N = p^i$ , 其中  $p$  是素数,  $i$  是一个整数, 那么  $J_N(1)$  中的所有元素都是二次剩余。令人欣慰的是, 一个素数的幂很容易分解(回顾习题 8.7 和习题 8.8 的提示)。

协议 18.4 允许 Alice 进行完备 ZK 证明  $E_{2\_Prime}$  中的成员归属。

---

**协议 18.4** ZK 证明  $N$  有两个不同的素因子

公共输入: 合数  $N$ ;

Alice 的秘密知识:  $N$  的分解;

对 Bob 输出:  $N \in E_{2\_Prime}$ 。



1. Bob 检验  $N$  不是素数或素数的幂(例如,运用 Prime\_Test 排除素数,并用习题 8.7 中的提示分解一个素数的幂);
2. Bob 在  $J_N(1)$  中选取  $m$  个随机数作为一个 Challenge 集合,并发送 Challenge 集合给 Alice。
3. 用  $x_1, x_2, \dots, x_k$  表示它们的平方都属于 Challenge; Alice 运用协议 18.3 向 Bob 证明这  $k$  个元素属于  $QR_N$ 。
4. 如果  $k > \lfloor \frac{3}{8}m \rfloor$ , Bob 接受证明; 否则拒绝。  
( \* 这里  $k > \lfloor \frac{3}{8}m \rfloor$  是一种“实用的少数判别准则”,见 4.4.1.2 节,那里讨论了“多数判别准则” $k > \frac{1}{2}m$ ; 因为  $QR_N$  中的元素在  $J_N(1)$  中不占多数,本协议不能简单套用那个准则; 在 18.5.1.2 节中将解释为什么要选用这个“判别准则” \* )

现在来考察协议 18.4 的安全性。

### 18.5.1.1 安全性

首先,显然协议 18.4 的完备零知识性直接来自于协议 18.3 的完备零知识性。下面只分析完全性和正确性。

#### 完全性

假设 Alice 诚实地构造了  $N \in E_{2\_Prime}$ ,但在运行协议后 Bob 仍然可能拒绝。这是因为碰巧 Bob 选取的随机提问中只有少于  $\frac{3}{8}$  的询问为平方数(Alice 运气不好!)。当完全性概率  $\epsilon < 1$  时就可能发生这种情况。

到目前为止我们所看到的其他协议中,验证者不会容忍任何错误,即使多轮重复也不允许一个错误。这些协议都是单边差错协议:如果示证者没有欺骗,那么完全性概率满足  $\epsilon = 1$ ,因此验证者当然就不会容忍哪怕是一个错误。在这里的协议 18.4 中,由于  $\epsilon = 1$  这一事实(当 Alice 没有欺骗的时候,见事实 4),Bob 可能碰巧选取了超过一半的非二次剩余,他应当容忍一定的错误。但是,如果差错数超过了预设的标准,那么 Bob 就应当认为是 Alice 在欺骗并拒绝接受证明。

如果 Alice 没有欺骗却被拒绝了,就称发生了一个 BadLuckAlice 事件。给定预设标准, Bob 以此来决定,我们来估计 BadLuckAlice 的概率。我们已选取了  $k > \lfloor \frac{3}{8}m \rfloor$  作为标准,也就是说,如果 Bob 看到有  $\frac{3}{8}$  或更多的询问为二次剩余,他接受协议,否则拒绝。18.5.1.2 节将解释为什么要选择这个标准。

我们来估计重复  $m$  轮后的  $\epsilon(m)$ 。考虑完全性概率界的下述等价形式,它将事件 BadLuckAlice 的意思表达得更清楚:

$$\text{Prob}[\text{BadLuckAlice}] = \text{Prob}[\text{Bob 拒绝} \mid N \in E_{2\_Prime}] < 1 - \epsilon(m)$$

在  $m = \# \text{ Challenge} < \# J_N(1)$  的条件下,事件 BadLuckAlice 是  $m$  次 Bernoulli 试验之和(见 3.5.2 节),其中,对  $k \leq \lfloor \frac{3}{8}m \rfloor$  的所有情况,共有  $k$  次“成功”和  $m - k$  次“失败”。由于 Alice 已经构造了  $N \in E_{2\_Prime}$ ,对包含  $J_N(1)$  中随机元素的 Challenge,在每一次 Bernoulli 试验中

“成功”和“失败”的概率都是  $1/2$ 。应用 3.5.2 节中给出的二项分布函数的“左尾部”(注意要将违反 Bob 标准的  $k$  的所有情形求和,即对所有  $k \leq \lfloor \frac{3}{8}m \rfloor$ ),我们有

$$1 - \epsilon(m) = \text{Prob}[\text{BadLuckAlice}] = \sum_{k=0}^{\lfloor \frac{3}{8}m \rfloor} \binom{m}{k} \left(\frac{1}{2}\right)^k \left(\frac{1}{2}\right)^{m-k} = \sum_{k=0}^{\lfloor \frac{3}{8}m \rfloor} \binom{m}{k} \left(\frac{1}{2}\right)^m$$

这是二项分布函数的“左尾部”(见 3.5.2.1 节“左尾部”的含义),因为点  $\frac{3}{8}m$  在中心点  $\frac{1}{2}m$  的左侧。

为了使 BadLuckAlice 概率可忽略地小,我们需选取  $m = 2000$ (理由在 18.5.1.2 节中给出)。这时“左尾部”为下面的值:

$$1 - \epsilon(2000) \approx 1.688 \cdot 10^{-29}$$

故  $\epsilon(2000)$  是一个压倒性概率。因此,如果 Alice 没有欺骗,则 Bob 会以一个压倒性概率接受证明。

根据大数定律(见 3.5.3 节),Bob 选取的询问次数越多,完全性概率就越高。顺便提一下,如果 Bob 选取  $\#J_N(1)$  个询问(尽管不切实际),完全性概率就变成了 1,即不会发生 Bob 一侧的差错(BadLuckAlice)。

### 正确性

对另一侧的错误,我们假设 Alice 不诚实地构造了  $N \notin E_{2\_Prime}$ (即  $N$  有多于两个的不同素因子),Bob 仍然有可能接受 Alice 的“证明”。这是因为碰巧 Bob 选取的随机询问中超过的提问为二次剩余(Bob 运气不好!)

用 BadLuckBob 表示条件事件  $N \notin E_{2\_Prime}$  而 Bob 接受了证明。对随机选取的 Challenge,我们从事实 5 知道,现在一个 Bernoulli 试验成功的概率至多为  $\delta = \frac{1}{4}$ ,失败的概率至少为  $1 - \delta = \frac{3}{4}$ 。应用二项分布公式对所有  $k > \lfloor \frac{3}{8}m \rfloor$  使 Bob 接受证明的情形求和,我们得到  $\delta(m)$  如下(二项分布函数的“右尾部”):

$$\delta(m) = \text{Prob}[\text{BadLuckBob}] = \sum_{k=\lfloor \frac{3}{8}m \rfloor}^m \binom{m}{k} \left(\frac{1}{4}\right)^k \left(\frac{3}{4}\right)^{m-k}$$

对  $m = 2000$ ,我们有

$$\delta(2000) \approx 1.847 \times 10^{-35}$$

Alice 还试图欺骗并幻想不被发觉就很愚蠢了!

到这里为止,已经全部考察了协议 18.4 的 ZK 性、完全性和正确性。

### 18.5.1.2 选择“判别准则”

当 Alice 没有欺骗的时候,一轮交互的完全性概率满足  $\epsilon = 1/2$ ,即  $J_N(1)$  中恰好有一半的元素为二次剩余,所以协议 18.4 不能用 4.4.1.1 节给出的“多数判别准则”来放大完全性概率。我们选择的标准  $3/8$  是  $\epsilon = 1/2$ (Alice 没有欺骗)和  $\delta = 1/4$ (Alice 欺骗了)的中点。这一选择使得两种“坏运气”事件差不多同样(不)可能。

这是一种“少数判别准则”。根据大数定律(见 3.5.3 节),只要  $\delta < \epsilon$ ,我们就能够选取它们的中点,重复多轮( $m$  轮)来缩小  $\delta(m)$  并放大  $\epsilon(m)$ 。因此重复充分多轮后就能够区分 Alice 欺骗和诚实,并且正确判断的可信度很高。

根据“经验法则”,为了使两种“坏运气”事件的概率都可以忽略,通常考虑为  $2^{-100}$  (到目前为止本章的所有协议一直坚持了这一法则),我们需要用 2000 作为重复次数。如果  $m$  从 2000 明显地减少,那么两种差错概率界都将急剧下降。例如,取  $m = 100$  (再根据“经验法则”,通常认为这是一个“可以接受”的重复次数),那么将得到  $\epsilon(100) \approx 0.993$  (这时 *BadLuckAlice* 发生的概率为  $1 - \epsilon(100) \approx 0.007$ ),  $\delta(100) \approx 0.0052$  (*BadLuckAlice* 发生的概率)。这些差错概率界远远不能令人满意,因为两种“坏运气”事件都太可能了(即两种“坏运气”事件的概率太显著了)。

通常,当  $\epsilon$  和  $\delta$  很接近时,双边差错协议不是很有效。

一些作者提出了更有效的单边差错( $\epsilon = 1$ ) ZK 协议证明  $N$  有两个素因子,例如, van de Graaf 和 Peralta[293]、Camenisch 和 Michels[64]、Gennaro、Miccianicio 和 Rabin[122]。这里介绍的协议基于 Berger、Kannan 和 Peralta 所提出的协议[33],在概念上是最简单的。双边差错特征在 ZK 协议中是很少见的,这是选择介绍本协议的另外一个重要原因,;因此我们想让读者对此有一些了解。

## 18.6 轮效率

现在考虑 18.1 节中列出的问题 II: 示证者要让验证者确信需要多少次交互? 这就是所谓的轮效率问题。一轮就是一个消息发送和接收活动的完整过程。由于许多 ZK (和 IP) 协议一般涉及到 Commit ( $P$  最先的一个步骤, 发起), Challenge ( $V$  的一个步骤), Response ( $P$  的第二个步骤, 应答), 我们通常称这样的三个步骤为一轮。

正如我们已经看到的一样, 一般来说, ZK 协议可以通过顺序重复多轮来降低差错概率。完全性概率  $\epsilon$  是式 (18.2.2) 的概率下界, 我们把  $1 - \epsilon$  则看做是差错概率上界。和正确性的情形一样, 这种差错概率界(上界)应当尽可能小。为了客观地衡量一个 ZK 协议的轮效率, 我们应当考虑由单轮获得的差错概率。单轮差错概率越小, 协议的轮效率越高。

粗略地说, 有三种不同规模的单轮差错概率, 它将协议也分为了三类不同轮效率的协议。

**对数轮协议** 迄今为止, 我们学过的所有 ZK 协议除了协议 18.4 外, 在单轮中都有常量差错概率, 如  $1/2$  或  $\log_2 \log_2 n$  (在诸如协议 18.1 或 Schnorr 身份识别协议中, 由于  $\log_2 n$  是一个安全参数, 我们将  $\log \log n$  等同为一个常量)。为了使差错概率降低到一个可忽略的小量, 即对所有的常数  $c$ , 降为以  $1/(\log n)^c$  为界的一个量, 具有常量差错概率的协议就必须重复  $\log n$  轮。因此这种协议称为对数轮协议。

**多项式轮协议** 对数轮协议的轮效率实际上是由关于安全参数的一个线性多项式度量的。有些 ZK 协议度量它们的轮效率要用次数高一些的多项式。通过通用多项式归约到 NPC 问题的任意一种 NP 语言的 ZK 协议(见 18.2.3 节)称为多项式轮协议。

协议 18.4 是一个多项式轮协议。首先, 由于其双边差错特性, 它需要更多的轮数。其次, 协议 18.4 在每一轮调用另外一个对数轮协议(见协议 18.3)。

**常数轮(或单轮)协议** 如果一个 ZK 协议在少许常数轮(或单轮)中就能够获得可忽略的小差错概率, 那么就没有必要重复对数多轮了。因此这种协议称为常数轮(或单轮)协议。

许多研究致力于提高 ZK 协议的轮效率, 已经得到很多结果。现在我们来看子群成员归属和离散对数问题的两个结果。

- 对  $\mathbb{Z}_N^*$  的子群, 其中  $N$  是奇合数, 18.6.1 节将推导其子群成员归属的 ZK 论据的轮效率下界。这是一个负面结果, 因为该下界是对数轮的, 即对该成员归属证明不存在常数轮协议。
- 对有限域  $\mathbb{F}_p$  中的元素, 将在 18.6.2 节中学习一个常数轮协议, ZK 证明离散对数相等。这是一个正面结果, 明显提高了 Schnorr 身份识别协议(协议 18.2)的轮效率。

### 18.6.1 子群成员归属的轮效率下界

让我们来再次考察协议 18.1 处理的子群成员归属(论据)问题。现在  $f(x)$  是 18.3.3.1 节中实现的情形; 即

$$f(x) = y \equiv g^x \pmod{N}$$

其中  $N$  是一个大的奇合数,  $g \in \mathbb{Z}_N^*$  有高的乘法阶。在这种实现中, 我们知道

$$\{g^x \pmod{N} \mid x \in \phi(N)\} \subset \mathbb{Z}_N^*$$

也就是说, 子集中的元素少于  $\phi(N)$ , 这是因为  $\mathbb{Z}_N^*$  不是循环的。

现在我们假设示证者 Alice 知道  $N$  的分解(回忆一下, 在 18.3.3 节中, 我们不允许 Alice 知道  $N$  的分解, 因此那里的变形协议是计算 ZK 的)。知道  $N$  的分解允许 Alice 进行完备 ZK 证明  $y \in \langle g \rangle$ 。

现在我们要问: 对  $f(x) = g^x \pmod{N}$ , 其中 Alice 知道合数  $N$  的分解, 像在 Schnorr 身份识别协议中做的那样, 通过增加 Bob 的询问长度能够提高协议 18.1 的轮效率吗?

回忆一下, 例如在 Schnorr 身份识别协议(协议 18.2)中, 我们稍微放大了询问:  $\text{Challenge} \in \{0, 1\}^{\log_2 \log_2 p}$ 。结果协议的变形提高了性能:  $\frac{m}{\log_2 \log_2 p}$  轮就够了, 而不是协议 18.1 需要的  $m$  轮, 同时正确性差错概率保持不变。

遗憾的是, 如果 Alice 知道  $N$  的分解, 那么就不可能再使用这种询问放大方法来提高轮效率。问题不在于 ZK 性质, 而在于正确性差错概率。无论用多大的询问, 该协议的正确性差错概率下界都是  $\delta = 1/2$ 。由于常量的和显著的正确性差错概率, 该协议必然是对数轮的。Galbraith、Mao 和 Paterson[119]注意到了现在就要揭示的事实。

为了解释清楚, 我们来考察使用扩展询问的单轮三次传输协议的正确性概率(因此, 正如已经在 18.3.2 节中学习的一样, 该协议是验证者诚实的 ZK 协议)。我们将看到, 考察的结果适用于多于一比特的任何询问长度。

这里我们描述一个验证者诚实的零知识协议证明  $\mathbb{Z}_N^*$  中的子群成员归属, 命名为“不要用”(协议 18.5)。必须提醒读者, 协议 18.5 的目的不是为了任何应用: 描述它只是为了揭示问题。

#### 协议 18.5 “不要用”

公共输入  $N$ : 一个大的奇合数;

$g, y: \mathbb{Z}_N^*$  中的两个元素, 满足  $g$  模  $N$  有高的阶;  $y \equiv g^z \pmod{N}$

Alice 的秘密输入: 整数  $z < \phi(N)$ ;

对 Bob 的输出:  $y \in \langle g \rangle$ , 即对某个  $z$ ,  $y \equiv g^z \pmod{N}$ 。

1. Alice 选择  $k \in_U \mathbb{Z}_{\phi(N)}$ , 计算  $\text{Commit} \leftarrow g^k \pmod{N}$ ; 发送 Commit 给 Bob;

2. Bob 均匀地选取随机  $\text{Challenge} < N$  并发送给 Alice;
3. Alice 计算  $\text{Response} \leftarrow k + z \cdot \text{Challenge} \pmod{\phi(N)}$ ;
4. 如果  $g^{\text{Response}} \equiv \text{Commit } y^{\text{Challenge}} \pmod{N}$ , Bob 接受证明; 否则拒绝。

表面看来,好像因为协议 18.5 的 Challenge 很大, Alice 难以猜测它, 因此她只好遵循协议指令, 随之将得到  $\delta \approx 1/\phi(N)$  数量级的正确性概率。如果真是这样, 那么该协议事实上就是单轮的。遗憾的是, 这个正确性概率估计不正确。例 18.4 演示了一种欺骗方法。

**例 18.4** 从现在起, 我们用记号  $\tilde{\text{Alice}}$ , 因为她下面所做的是不诚实的。

由于知道  $N$  的分解,  $\tilde{\text{Alice}}$  很容易计算 1 的一个非平凡平方根, 即元素  $\xi \in \mathbb{Z}_N^*$ , 满足  $\xi \neq \pm 1$ , 且  $\xi^2 \equiv 1 \pmod{N}$ 。可以用算法 6.5 计算平方根。她可以选取  $\xi$  使得  $\xi \notin \langle g \rangle$ 。

现在,  $\tilde{\text{Alice}}$  计算公共输入为

$$Y \leftarrow \xi g^z \pmod{N}$$

显然,  $Y \in \xi \langle g \rangle$ , 即  $Y$  属于  $\langle g \rangle$  的陪集。我们明显注意到  $Y \notin \langle g \rangle$ , 因为  $\xi \notin \langle g \rangle$  (见定理 5.1 证明中陪集的性质, 5.2.1 节)。

$\tilde{\text{Alice}}$  不是按照协议指令计算 Commit, 而是在猜测 Bob 询问的奇偶性时掷一枚均匀的硬币  $b \in_U \{0, 1\}$ 。然后她如下计算 Commit:

$$\text{Commit} \leftarrow \begin{cases} g^k \pmod{N} & \text{如果 } b = 0 \\ \xi g^k \pmod{N} & \text{如果 } b = 1 \end{cases}$$

在协议余下的部分,  $\tilde{\text{Alice}}$  像协议描述指示的那样继续下去。

显然,  $\tilde{\text{Alice}}$  有  $1/2$  的机会猜测正确。当正确猜测到偶数  $\text{Challenge} = 2u$  时, Bob 的验证步骤为:

$$g^{\text{Response}} \equiv g^k g^{z \cdot 2u} \equiv \text{Commit}(\xi g)^{z \cdot 2u} \equiv \text{Commit } Y^{\text{Challenge}} \pmod{N}$$

所以 Bob 将接受证明。当正确猜测到奇数  $\text{Challenge} = 2u + 1$  时, Bob 的验证步骤为:

$$g^{\text{Response}} \equiv g^k g^{z(2u+1)} \equiv \xi g^k (\xi g)^{z(2u+1)} \equiv \text{Commit } Y^{\text{Challenge}} \pmod{N}$$

所以 Bob 也将接受证明。

因此, 无论 Bob 的提问多大, 我们都只能得到协议 18.5 的单轮正确性概率为  $\delta = 1/2$ 。这就是为什么我们命名该协议为“不要用”的原因。  $\square$

由于 Bob 不知道  $N$  的分解, 所以他不能独自确定子群成员归属 (见注释 18.1 及其随后的困难性讨论)。因此, 除了正确性概率为  $1/2$  之外, Bob 无法阻止运用例 18.4 给出的方法进行欺骗。扩大询问的规模不会提供任何帮助。

注意到例 18.4 中的问题在 18.3.3.2 节的 (计算 ZK) 协议中并不出现, 那里用类似的方式实现了  $f(x)$ , 例如  $f(x) = a^x \pmod{N}$ , 其中  $N$  是奇合数。回忆一下, 那个协议使用比特询问, 因此它的正确性差错概率是同样的值  $\delta = 1/2$ 。也要注意 Schnorr 身份识别协议对这个问题是免疫的, 因为该协议中的群  $\langle g \rangle$  阶为素数  $q$ , 除了单位元, 不包含任何阶小于  $q$  的元素。

运用 1 模  $N$  的非平凡平方根给  $\tilde{\text{Alice}}$  提供了  $\delta = 1/2$  的成功欺骗概率。用平凡的情形  $\xi = -1$  (另外一种平凡情形  $\xi = 1$  不构成攻击) 似乎使得 Bob 更深信不疑:  $Y$  或  $-Y$  都属于  $\langle g \rangle$ 。

但是,由于 Alice 知道  $N$  的分解而 Bob 不知道,她可以用其他低阶乘数盲化  $g^k$ , 例如一个 3 阶乘数,这可以用中国剩余定理(6.2.3 节的定理 6.7,用中国剩余定理,可以计算任何阶为  $d|\phi(N)$  的元素)。因此,正确性差错概率不可能是可忽略的量。对一般的安全参数设置,包括  $\mathbb{Z}_N^*$  的子群情形,协议 18.1 仍然是证明(ZK 论据)子群成员归属问题的惟一形式。

到此为止,我们得出结论,ZK 子群成员归属一般情况下是一个对数轮问题。

在下一章要介绍的一个 ZK 协议应用中,需要证明  $\mathbb{Z}_N^*$  中的子群成员归属。不过在那个应用中,我们不能承受对数轮协议的开销,那里将利用对  $N$  的一个特殊设置来绕开这一问题。

### 18.6.2 离散对数的常数轮证明

Schnorr 身份识别协议(协议 18.2)提供拥有有限域  $\mathbb{F}_p$  中元素的离散对数的 ZK 论据。我们已经知道它是对数轮协议。

现在证明,对于 Schnorr 身份识别协议解决的同样问题,可以得到常数轮效率的 ZK 证明,这要归功于 Chaum 的一个协议[73]。称这个协议为 **Chaum 的 ZK 离散对数相等性证明协议**,它 ZK 证明两个元素有同样的离散对数值。

我们用与 Schnorr 身份识别协议同样的安全参数设置介绍 Chaum 的 ZK 离散对数相等性证明协议。也就是说,设元素  $g \in \mathbb{F}_p$ ,  $p$  是一个奇素数,  $\text{ord}_p(g) = q$ , 其中  $q$  也是一个奇素数(因此  $q|p-1$ )。记  $G = \langle g \rangle$ 。

Chaum 的 ZK 离散对数相等性证明协议用到另外一个元素  $h \in \langle g \rangle$ , 其中  $h \neq g$  且  $h \neq 1$ 。协议 18.6 描述 Chaum 的协议。

从协议描述我们知道,该协议要交换四组消息,但只需执行一次。在正确性分析中将看到这个单轮协议获得正确性差错概率  $\delta = 1/q$ 。因此,Chaum 的 ZK 证明离散对数协议极为有效。

现在考察该协议的安全性。

#### 18.6.2.1 Chaum 的 ZK 证明离散对数协议的安全性

##### 完全性

直接观察协议就得到完全性概率  $\epsilon = 1$ , 也就是说,如果 Alice 拥有  $z$  并遵循了协议指令,那么 Bob 总是接受证明。

##### 正确性

我们将看到,Chaum 的 ZK 离散对数相等性协议是一个证明协议,也就是说,示证者 Alice 可以是一个计算上无界的参与方。为此,在正确性分析中对 Alice 的计算资源不做任何限制。

假设 Alice 欺骗,因此公共输入值  $(p, q, g, h, X, Y)$  满足下面离散对数不等性条件:

$$\text{对某些 } z \neq z' \pmod{q}, \text{ 有 } X \equiv g^z \pmod{p}, Y \equiv h^{z'} \pmod{p} \quad (18.6.1)$$

为了使 Bob 接受她的证明,也就是要在第 5 步中通过他的验证, Alice 必须在第 2 步发送  $\text{Commit}_A^{(2)}$  给 Bob, 满足

$$\text{Commit}_A^{(2)} \equiv X^c X^a Y^b \pmod{p} \quad (18.6.2)$$

换句话说, Alice 在从 Bob 那里接收到  $a, b$  之后,必须解开她的承诺值  $c \in \mathbb{Z}_q$ , 满足式(18.6.2)。由于在第 1 步 Bob 给定了  $a, b$ , Alice 在第 2 步给定了  $\text{Commit}_A^{(1)}, \text{Commit}_A^{(2)}$  所以式(18.6.2)是说  $c \in \mathbb{Z}_q$  也在第 2 步给定了。换句话说, Alice 在第 2 步发送她的承诺之后不能改变  $c$ 。



由于  $c \in \mathbb{Z}_q$  也在第 2 步给定了, 我们有

$$c \equiv \log_g \frac{\text{Commit}_A^{(1)}}{g^a h^b} \pmod{q} \quad (18.6.3)$$

根据式(18.6.2), 我们还有

$$c \log_g X \equiv \log_g \frac{\text{Commit}_A^{(2)}}{X^a Y^b} \pmod{q} \quad (18.6.4)$$

由于  $h \in \langle g \rangle$  (因为  $\text{ord}_p(h) = q$ , Bob 可以通过检验  $h \neq 1$  且  $h^q \equiv 1 \pmod{p}$  来证实这一点), 对某个  $d \in \mathbb{Z}_q, d \neq 0 \pmod{q}$ , 我们可以记  $h \equiv g^d \pmod{p}$ , 从而式(18.6.3)可以重写成下面的等价形式:

$$c - \log_g \text{Commit}_A^{(1)} \equiv -a - bd \pmod{q} \quad (18.6.5)$$

类似地运用式(18.6.1), 可以将式(18.6.4)重写为

$$c \log_g X - \log_g \text{Commit}_A^{(2)} \equiv -az - bdz' \pmod{q} \quad (18.6.6)$$

由于  $z \neq z' \pmod{q}$ , 式(18.6.5)和式(18.6.6)构成了下列线性同余方程组:

$$\begin{pmatrix} c - \log_g \text{Commit}_A^{(1)} \\ c \log_g X - \log_g \text{Commit}_A^{(2)} \end{pmatrix} = \begin{pmatrix} -1 & -d \\ -z & -dz' \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} \pmod{q}$$

这个线性同余方程组的矩阵是满秩的(秩 = 2)。由线性代数的一个简单事实, 该方程组有惟一解  $(a, b) \in \mathbb{Z}_q \times \mathbb{Z}_q$ 。这个解满足 Bob 第 1 步构造的  $\text{Commit}_B$  和他第 5 步的验证。

但是, 在第 2 步, 当 Alice 固定  $c \in \mathbb{Z}_q$  时, 她还只得到一个方程(18.6.5)。她根据该方程恰好有  $q$  个不同的  $(a, b)$  对。这  $q$  个对均满足式(18.6.5), 但只有一个还满足式(18.6.6), 这是 Bob 第 5 步的验证。因此, 即使计算上是无限的, Alice 在第 2 步指出正确的  $(a, b)$  对的概率也正好为  $1/q$ 。

到这里, 我们不仅得到  $1/q$  是 Chaum 协议单轮的正确性差错概率, 而且还得到该协议提供了离散对数相等性的一个证明(即不只是一个论据)。

### 完备零知识性

最后, 我们来考察协议 18.6 的 ZK 性质。

### 协议 18.6 Chaum 的 ZK 离散对数相等性证明协议

公共输入:

$p, q$ : 两个素数, 满足  $q \mid p-1$ ;

(\* 典型的长度设置:  $|p| = 1024, |q| = 160$  \*);

$g, h$ :  $\text{ord}_p(g) = \text{ord}_p(h) = q, g \neq h$ ;

(\* Bob 检验:  $g \neq 1, h \neq 1, g \neq h, g^q \equiv h^q \equiv 1 \pmod{p}$  \*);

$X, Y$ :  $X = g^z \pmod{p}, Y = h^z \pmod{p}$ ;

Alice 的秘密输入:  $z \in \mathbb{Z}_q$ ;

对 Bob 的输出: Alice 知道某个  $z \in \mathbb{Z}_q$ , 满足  $X \equiv g^z \pmod{p}$  且  $Y \equiv h^z \pmod{p}$ , 或  $\log_g X \equiv \log_h Y \pmod{q}$ 。

1. Bob 选取  $a, b \in_U \mathbb{Z}_q$ , 计算  $\text{Commit}_B \leftarrow g^a h^b \pmod{p}$ ; 发送  $\text{Commit}_B$  给 Alice;  
(\*  $\text{Commit}_B$  是 Bob 的询问 \*)
2. Alice 选取  $c \in_U \mathbb{Z}_q$ ; 她计算  
 $\text{Commit}_A^{(1)} \leftarrow \text{Commit}_B g^c \pmod{p}$ ,  $\text{Commit}_A^{(2)} \leftarrow (\text{Commit}_A^{(1)})^z \pmod{p}$ ;  
 她发送  $\text{Commit}_A^{(1)}$ ,  $\text{Commit}_A^{(2)}$  给 Bob;
3. Bob 向 Alice 公开  $a, b$ ;  
(\* Bob 解开他的承诺, 目的是证明他正确地构造了询问 \*)
4. Alice 验证是否有  $\text{Commit}_B \equiv g^a h^b \pmod{p}$ ; 如果等式成立, 她向 Bob 公开  $c$ , 否则中止协议;  
(\* 只有 Bob 正确地构造了询问, Alice 才解开承诺, Bob 正确地构造他的询问意味着他已经知道 Alice 要公开的  $X^a Y^b \pmod{p}$  \*)
5. Bob 验证  
 $\text{Commit}_A^{(1)} \equiv \text{Commit}_B g^c \pmod{p}$ ;  $\text{Commit}_A^{(2)} \equiv X^c X^a Y^b \pmod{p}$ ;  
 如果等式成立, 他接受证明, 否则拒绝。

该协议事实上是完备 ZK 的。让我们构造一个等同器  $\mathcal{EQ}$  来生成具有和证明副本同样分布的副本。对公共输入元组  $(p, q, g, h, X, Y)$ ,  $\mathcal{EQ}$  执行下列简单有效的步骤:

1.  $\mathcal{EQ}$  选取  $a, b \in_U \mathbb{Z}_q$ , 计算  $\text{Commit}_B \leftarrow g^a h^b \pmod{p}$ ;
2.  $\mathcal{EQ}$  选取  $c \in_U \mathbb{Z}_q$ , 计算  
 $\text{Commit}_A^{(1)} \leftarrow \text{Commit}_B g^c \pmod{p}$ ,  
 $\text{Commit}_A^{(2)} \leftarrow X^c X^a Y^b \pmod{p}$ ;
3.  $\mathcal{EQ}$  输出  $\text{Transcript} = \text{Commit}_B, \text{Commit}_A^{(1)}, \text{Commit}_A^{(2)}, a, b, c$ 。

很容易验证这个 Transcript 和证明副本具有同样的分布。

还有一种不同但更令人信服的方式证明 Chaum 协议的完备 ZK 性。首先, 如果 Bob 想蒙骗发送出一个不合法询问, 即  $\text{Commit}_B$  不是正常构造的, 那么他将什么也收不到。其次, 如果 Bob 确实正确发送了用  $(a, b) \in \mathbb{Z}_q \times \mathbb{Z}_q$  构造的询问, 那么他在第 1 步一开始就已经知道了 Alice 要“公开”的值  $X^a Y^b \pmod{p}$ 。在两种情形中, Bob 均完全没有得到关于 Alice 秘密输入新的信息!

#### 18.6.2.2 讨论

- Chaum 的 ZK 离散对数相等性协议可以用做身份识别协议。在这种应用中, 元素对  $(g, X)$  可以是用户的公钥材料, 由一个公钥证书机构发放证书 (CA, 见 13.2 节)。
- 计算  $g^a h^b \pmod{p}$  和  $X^c X^a Y^b \pmod{p}$  可以用算法 15.2, 获得和计算单一模指数近似的开销。因此 Alice 和 Bob 的开销都大约为三次模指数。以此开销, 证明获得了对 Alice 欺骗可忽略小的差错概率。与此相对比, 为了获得近似小的差错概率, Schorr 身份识别协议要求 Alice 和 Bob 计算  $\log_2 p \approx 10$  (在  $p \approx 2^{1024}$  的情况下) 次模指数运算。

- 示证者无计算资源限制使得该协议可以用于示证者是很强大的参与方时的应用,如政府机构。
- 尽管正确性证明是一个有力的证明,但它并不表明 Alice 需要知道离散对数值。它所表明的是她用正确的指数回答了提问,也许她把某些人当做指数预言机来用。在 Schnorr 身份识别协议中,两个正确回答即使是示证者从预言机那里得到的,也构成一个提取离散对数的知识提取器,这正是证明三元组 ElGamal 签名(见 16.3.2 节)的分叉引理技术的基础。在这里的 Chaum 协议中,两个正确的回答并不构成离散对数值的知识提取器。
- Chaum 是为不可否认签名方案[73](也在 Chaum 和 Antwerpen[75]中)提出的这个协议。“不可否认签名方案”提供文件作者身份的一个证明,用交互式协议代替普通签名方案中的签名验证程序。因此,它使得签名者能够选择签名验证者,从而保护了签名者签名的隐私权。这在不希望可公开验证签名的某些应用中可能有用。例如,一个软件销售商在它的产品上加了数字签名,使得它可以鉴别它的产品为原版,没有病毒,但只想让付费用户才能够验证这些签名的有效性。运用不可否认签名,销售商可以防止盗版者使其他人相信盗版软件的质量。

## 18.7 非交互式零知识

我们已经看到,作为交互式协议,ZK 协议一般要求交互。尽管对于单轮或常数轮协议(例如 Chaum 的 ZK 证明离散对数相等性协议)的情形,交互的次数很少,但需要交互意味着示证者和验证者都必须同时在线。如果一个 ZK 证明(或论据)可以做到无须交互,那么就可以用“单向”通信方式。这种通信方式有不少优点。

考虑一个想像的例子。 $P, V$  是数学家([45]中设想的一个情景),前者正在周游世界,在这期间发现了新的数学定理证明,他想用 ZK 向后者证明这些新定理。在这种场合,非交互式证明是必要的,因为  $P$  很可能没有固定的地址,在任何邮件到达之前就离开了。这两个想像中的用户会很喜欢非交互式 ZK 证明。

在第 15 章一开始我们就讨论了非交互式 ZK 证明更现实的应用:构造对 CCA2 攻击者可证明安全的公钥加密方案(尽管我们介绍第 15 章的目的只是对这种保证加密方案安全的方法的一个建议)。无论如何,可能构造非交互式 ZK 证明(或论据)总是一个有用的额外特征。

如果  $P, V$  共享随机询问比特,Blum、Feldman 和 Micali 提出了一种获得非交互式 ZK (NIZK)的方法[45]。共享的随机询问比特可以由一个  $P$  和  $V$  都信任的第三方提供(这样一个相互信任的随机源 Rabin 称之为随机信标[241],“天上掉下来的随机性”),也可能是双方在一起时(例如,在那个想像出来的数学家匆匆离开,去周游世界之前)就已生成。

在 18.3.2.2 节我们就介绍了 Fiat-Shamir 启发式作为一种构造非交互式“知识证明”<sup>①</sup>的一般方法。但是由 Fiat-Shamir 启发式获得非交互性的代价是失去了 ZK 性质:“暗中证明”变成了“公开的”,即变成了公开可验证的。

Jakobsson、Sako 和 Impagliazzo 设计了一种有趣的技术,它使用 Fiat-Shamir 启发式却保持了

<sup>①</sup> 我们总是对源于 Fiat-Shamir 启发式的术语“知识证明”用引号的形式,因为严格来说,它是一个知识论据,见 18.4.1 节。

“在暗中证明”的性质[155]。他们称之为指定验证者证明:如果 Alice 进行了一个要 Bob 验证的证明,那么只有 Bob 可以确信证明的有效性。在其他任何人看来该证明或者是 Alice 进行的,或者是 Bob 仿真的。

### 18.7.1 利用指定验证者获得 NIZK

通过 Alice 对下述逻辑表达式构造一个源于 Fiat-Shamir 启发式的非交互式“知识证明”,这就是 Jakobsson 等的 NIZK 技术:

“Alice 的断言为真” $\vee$ “Bob 仿真了 Alice 的证明”

由于一种称为陷门承诺的原型(也被 Brassard、Chaum 和 Crépeau 称为可仿真承诺 [60]), Alice 能够实现“证明”这一逻辑表达式。

陷门承诺是 Alice 用指定验证者 Bob 的公钥构造的一种特殊承诺。我们用

$$\text{TC}(w, r, y_B)$$

表示用 Bob 的公钥  $y_B$  构造的一个陷门承诺。在该承诺中,  $w$  是承诺的值(由构造它的主体承诺),  $r$  是一个随机输入。性质 18.1 指出了  $\text{TC}(w, r, y_B)$  的两个重要性质。

#### 性质 18.1 陷门承诺性质

- i) 如果不知道  $y_B$  的秘密部分, 承诺是绑定的, 即不存在有效算法计算一对碰撞  $w_1 \neq w_2$ , 使得  $\text{TC}(w_1, r, y_B) = \text{TC}(w_2, r', y_B)$ 。
- ii) 利用  $y_B$  的秘密部分, 很容易计算任意数量的碰撞对。

**例 18.5 一个陷门承诺方案** 设  $(p, q, g)$  是 Schnorr 身份识别协议的公共输入,  $y_B = g^{x_B} \pmod{p}$  是 Bob 的公钥, 其中  $x_B \in \mathbb{Z}_q$  是他的秘密指数。

如果 Alice 想承诺值  $w \in \mathbb{Z}_q$ , 她选取  $r \in_U \mathbb{Z}_q$ , 计算  $\text{TC}(w, r, y_B) \leftarrow g^w y_B^r \pmod{p}$ 。她可以通过展示对  $(w, r)$  公开(解开承诺)  $\text{TC}(w, r, y_B)$ 。我们现在来证明  $\text{TC}(w, r, y_B)$  满足陷门承诺的两条性质。

证明 TC 性质 (i): 如果不知道 Bob 的私钥  $x_B$ ,  $(w, r)$  是 Alice 解开承诺的惟一方式。假设与此相反, 她还知道一对不同的承诺值  $(w', r')$ ,  $w' \neq w \pmod{q}$  (因此  $r' \neq r \pmod{q}$ )。那么由于

$$1 = g^{w-w'} y_B^{r'-r} \pmod{p}$$

我们得到

$$y_B \equiv g^{x_B} \equiv g^{\frac{w'-w}{r-r'}} \pmod{p}$$

即 Alice 知道  $x_B \equiv \frac{w'-w}{r-r'} \pmod{q}$ , 这与假设 Alice 不知道  $x_B$  矛盾。

证明 TC 性质 (ii): 利用  $x_B$ , Bob 可以选取  $w_1, w_2, r_1 \in_U \mathbb{Z}_q$ , 其中  $w_1 \neq w_2 \pmod{q}$ 。然后他设置

$$r_2 \leftarrow \frac{w_1 - w_2 + r_1 x_B}{x_B}$$

直接可以验证  $\text{TC}(w_1, r_1, y_B) = \text{TC}(w_2, r_2, y_B)$ 。

□

在 18.3.2.2 节我们已经看到,由 Fiat-Shamir 启发式得到的“知识证明”是由 Alice 构造的一个三元组 (Commit, Challenge, Response), 在这个三元组中, Commit 是一个承诺, Alice 在其中承诺了一个值  $k$ , 一旦承诺后她不能再改变。

在 Jakobsson 等的 NIZK 方案中, 一个证明是下面的元组:

$$(w, r, \text{Commit}, \text{Challenge}, \text{Response}) \quad (18.7.1)$$

这里前缀对  $(w, r)$  是 Alice 对  $\text{TC}(w, r, y_B)$  解开的承诺, 附加这一元对的目的是为了让指定的验证者 Bob 用他的陷门信息找到碰撞。Bob 找到碰撞的能力使得他能够仿真 Alice 的证明。

### Alice 构造证明的程序

Alice 构造式 (18.7.1) 中的证明元组如下:

- P.1 选取  $w, r \in_U \mathbb{Z}_q$ , 计算  $\text{TC}(w, r, y_B) \leftarrow g^w y_B^r \pmod{p}$ ;
- P.2 按 Fiat-Shamir 启发式那样计算 Commit: 选取  $k \in_U \mathbb{Z}_q$ , 计算  $\text{Commit} \rightarrow g^k \pmod{p}$ ;
- P.3 通常那样生成 Challenge: 使用一个杂凑函数(可能将  $M$  也作为一个可选消息):  
 $\text{Challenge} \leftarrow h(\text{TC}(w, r, y_B) \parallel \text{Commit} \parallel [M]);$
- P.4 现在 Response 的计算也将承诺  $w$  作为输入:  $\text{Response} \leftarrow k + x_A (\text{Challenge} + w) \pmod{q}$ 。

### Bob 的验证程序

给定式 (18.7.1) 中的证明元组(可能包括可选消息  $M$ ), Bob 用下面的程序验证:

- V.1  $\text{Challenge} \leftarrow h(\text{TC}(w, r, y_B) \parallel \text{Commit} \parallel [M]);$
- V.2 检验  $g^{\text{Response}} \stackrel{?}{=} \text{Commit} y_A^{\text{Challenge}} y_A^w \pmod{p}$ ; 如果通过检验则接受证明, 否则拒绝。

下面我们来考察该方案的安全性。

#### 18.7.1.1 安全性

##### 完全性

在式 (18.7.1) 中, Alice 的证明元组与由 Fiat-Shamir 启发式生成的 (Commit, Challenge, Response) 情形很相似。“指定验证者证明”不同于 Fiat-Shamir 启发式证明的惟一元素是额外的值  $y_A^w \pmod{p}$ , 这一额外的值乘在 Bob 验证程序(步骤 V.2)表达式的右边。因此, 该方案的完全性是直接的。

##### 正确性

在指定的验证者 Bob 看来, 值  $y_A^w \pmod{p}$  是固定的, 因为其中的  $w$  固定在  $\text{TC}(w, r, y_B)$  中, 并由于 TC 性质(i), Alice 不能改变它, 除非她知道 Bob 的私钥  $x_B$ 。因此, 如果 Bob 确信 Alice 不知道他的私钥  $x_B$ , 则乘数  $y_A^w \pmod{p}$  是一个常数, 从而三元组 (Commit, Challenge, Response) 是一个基于 Fiat-Shamir 启发式论据, 由 Alice 生成。从而方案的正确性和由 Fiat-Shamir 启发式生成的论据一样。我们之所以这样说, 是因为 Alice 的计算资源必然是多项式有界的(使她不能对杂凑函数或 Bob 的公钥求逆)。本方案是一个论据。

##### 完备 ZK 性

在其他方看来, 由于 Bob 知道陷门信息  $x_B$ , 出现在 Bob 验证步骤(步骤 V.2)表达式右边的

乘数  $y_A^w \pmod p$  不再是固定的常数,它可以是 Bob 随意操控的任何值。实际上,由于 Bob 能够随意仿真  $\text{TC}(w, r, y_B)$ , 所以式(18.7.1)中的证明元组也可以完全仿真。现在来看这个仿真。

### Bob 的仿真程序

Bob 选取  $\text{Response}, \alpha, \beta \in_U \mathbb{Z}_q$ , 计算

- S.1  $\text{TC}(w, r, y_B) \leftarrow g^\alpha \pmod p$
- S.2  $\text{Commit} \leftarrow g^{\text{Response}} y_A^{-\beta} \pmod p$
- S.3  $\text{Challenge} \leftarrow h(\text{TC}(w, r, y_B) \parallel \text{Commit} \parallel [M])$
- S.4  $w \leftarrow \beta - \text{Challenge} \pmod q$
- S.5  $r \leftarrow (\alpha - w)/x_B \pmod q$
- S.6 他输出元组  $(w, r, \text{Commit}, \text{Challenge}, \text{Response})$  作为仿真证明。

可以证明这个仿真是完备的。

首先,由于步骤 S.2, 我们有

$$g^{\text{Response}} \equiv \text{Commit} y_A^\beta \pmod p$$

然后经过步骤 S.3, 右边变为

$$\text{Commit} y_A^\beta \equiv \text{Commit} y_A^{w + \text{Challenge}} \equiv \text{Commit} y_A^{\text{Challenge}} y_A^w \pmod p$$

也就是说, 我们得到希望的

$$g^{\text{Response}} \equiv \text{Commit} y_A^{\text{Challenge}} y_A^w \pmod p$$

它和步骤 V.2 的验证一致。

其次, 根据步骤 S.5, 我们有

$$g^w y_B^r \equiv g^{w + rx_B} \equiv g^\alpha \pmod p$$

检验步骤 S.1 中的  $\text{TC}(w, r, y_B)$  构造, 陷门承诺的构造确实是正确的。

最后, 很容易验证, 这些值不仅像证明的那样具有正确的构造, 而且和 Alice 生成的证明具有同样的分布。因此, Bob 的仿真算法是等同的。完备 ZK 性成立。

### 18.7.1.2 应用

Jakobsson 等为他们的“指定验证者证明”技术设想了有趣的应用。其一就是“不可否认签名”的一种高效实现(见在 18.6.2.2 节对“不可否认签名”的讨论): 我们描述中的可选消息  $M$  可以认为是 Alice 的签名, 该签名只能由指定的验证者 Bob 验证。考虑软件销售商认证其产品正宗性的一个应用: 如果销售商 Alice 运用“指定验证者证明”让购买者 Bob 验证, 那么 Bob 不能使第三方相信他所购买的拷贝的正宗性, 因为他能够仿真一个“指定验证者证明”。

另外一个很好的应用是电子投票。选举中心在收到投票者 Carol 的投票后, 必须发送给 Carol 一个收据, 使她相信已经正确地统计了她的投票。在这里, 选举中心让 Carol 相信中心证明的正确性是非常重要的, 必须防止武装胁迫者 Malice 强迫 Carol 投他要选的候选人。现在如果收据是利用“指定验证者证明”技术构造的, 那么 Malice 就不能检验正确性: 显然, Carol 完全可以对 Malice 选中的候选人仿真一份收据。这种安全服务称为无收据的电子投票, Benaloh 和 Tuinstra 对此进行了研究[31]。



## 18.8 本章小结

本章对零知识协议进行了研究。

我们首先介绍了交互式证明系统,并指出 IP 协议与在第 4 章学习的复杂性类  $NP$  密切相关,这使得我们可以更好地理解  $NP$  中的一些困难问题。在学完第 4 章之后,我们知道,对于一种语言  $L \in NP$ ,如果(不)存在一个带有论据的算法,问题  $I \in L$  是容易(困难)的。在学完本章以后,我们进一步以一种更加直观的方式知道,如果有(没有)示证者合作,同样的判定问题是容易(困难)的。

随后,我们给出了零知识性的几个概念:完备性、诚实验证者、计算的与统计的、证明和论据概念的区别,考虑了具有双边差错概率特性的协议,考察了轮效率问题,最后研究了非交互式零知识协议。在介绍以上每一种概念时,我们都给出了实用的协议来具体说明。尽管零知识协议被视为一种高级的密码学课题,但我们希望通过这种学习方式,可以让那些想开发能提供新奇而实用服务的信息安全系统的读者易于理解。

零知识协议在密码学中是一个活跃的研究领域(与理论计算机科学有联系)。对那些想进一步研究本课题的读者,这一章对概念做了基础性的介绍,它们对理解那些尚未介绍的高级研究文章是必要的。

## 习题

### 18.1 解释零知识(ZK)协议中的下列概念:

- i) 公共输入
- ii) 秘密输入
- iii) 随机输入
- iv) 完全性
- v) 正确性
- vi) 证明副本
- vii) 欺骗示证者
- viii) 不诚实验证者
- ix) 等同性
- x) 可仿真性

### 18.2 区分下列概念:

- i) 完备零知识
- ii) 诚实验证者的零知识
- iii) 计算零知识
- iv) 统计零知识
- v) 零知识证明
- vi) 零知识论据
- vii) 知识证明

- 18.3 由数字签名提供的不可否认服务意味着如下的一个知识证明:签名者有一个私钥(知识),使他(她)能发布签名。这种意义上的知识证明和零知识证明有何区别?
- 18.4 零知识协议能否是一个零知识论据? 计算零知识协议能否是一个零知识证明?
- 18.5 在零知识协议中,示证者的计算能力必须是多项式有界的吗? 请对于验证者的情况也回答同样的问题。
- 18.6 为什么 Schnorr 身份识别协议不是常数轮的?
- 18.7 证明 Schnorr 身份识别协议(协议 18.2)的完全性。
- 18.8 在 18.3.3.2 节描述的计算零知识协议中,我们已经讨论了 Alice 可以从集合  $\mathbb{Z}_{N^{1+\alpha}}$  中选择任意小的固定  $\alpha > 0$  生成承诺,为什么?
- 18.9 证明 18.4.1 节中的事实 3。
- 18.10 某些 ZK 协议要使用多轮来减少(正确性)差错概率。通常验证者只有在所有轮都没有检测到错误的情况下才接受证明,这种“判别准则”能否用于双边差错协议?
- 18.11 为什么协议 18.4 不能用“多数判别准则”?
- 18.12 为什么双边差错协议效率不高,尤其是在单轮消息交换中诚实示证者和欺骗示证者的表现相似的情况下更是如此?
- 18.13 什么是常数(对数,多项式)轮协议?
- 18.14 协议 18.6 能否简化成诚实验证者的三次传输形式?  
提示:如果 Alice 对 Bob 的询问直接计算模指数,那么第 2 步、3 步和第 4 步可以压缩成一次消息传送。
- 18.15 协议 18.6 在上一个问题中建议的诚实验证者的形式有什么不安全的地方?  
提示:参考 18.6.2.2 节中讨论的第 4 点。
- 18.16 什么是陷门承诺?
- 18.17 非交互 ZK 协议有哪些应用?

## 第 19 章 回到“电话掷币”协议

本书的第一个密码协议——“电话掷币”(协议 1.1)是用一个“奇妙函数” $f$ 来描述的。让我们再来考察  $f$  的两个性质(性质 1.1):

- I) 对任意整数  $x$ , 从  $x$  计算  $f(x)$  是容易的; 而给定任意值  $f(x)$ , 要得到它的一个原像  $x$  的任何信息是不可能的, 不管  $x$  是奇数还是偶数。
- II) 找到一个整数对  $(x, y)$ , 满足  $x \neq y$  但  $f(x) = f(y)$  是不可能的。

到目前为止, 这个“奇妙函数”仍然是奇妙的, 连“不可能”一词的支持证据都没有给出, 更不用说函数的具体实现了(对协议 1.1 也同样如此)。

事实上, 在 1.2.1 节中我们确实建议了一个实现协议 1.1 的实用方法: 用诸如 SHA-1 那样的实用杂凑函数来实现  $f$ 。在 SHA-1 的实现中, 对任意整数  $x$ ,  $f(x)$  可以编码成 40 位的 16 进制字符, 因此, Alice 把  $f(x)$  通过电话念给 Bob 听是切合实际的。我们还提到, 这个实现对两个朋友确定一个娱乐地点来说已经足够好了。

可是, 在很多密码学应用中, 互不信任的通信双方需要用到都信任的随机数。这种应用的安全后果要比玩一个开心游戏严肃得多。例如, 我们已经看到, 在整本书的许多攻击中, 其中的一种标准攻击技术, 说到底, 就是诱骗一个天真的用户提供预言机服务, 其中该用户对看起来无害的“随机”数执行一种密码运算。如果用户可以高度确信要处理的随机数确实是真的, 不管是否是来自预言机服务请求的, 那么很多此类攻击便不复存在。因此, 真随机性和知道一个看起来随机的数确实是随机的在密码系统安全中非常重要。

为了看清在需要可信随机源背后的另一个原因, 让我们回忆上一章诚实验证者的零知识协议。它需要彼此都信赖的随机询问, 这些随机询问不应该是从杂凑函数导出来的。不诚实验证者之所以能够攻击诚实验证者的零知识协议, 正是因为他(她)能使用杂凑函数来产生一个看起来“随机”的询问(见 18.3.2.1 节)。因此, 就像我们在第 1 章中建议的那样, 使用像 SHA-1 一类实用杂凑函数实现协议 1.1(生成彼此都信任的随机数而不只是确定一个娱乐地点的掷币协议), 当然对这类应用是不合适的。

利用实用杂凑函数实现掷币协议不合适的另外一个原因是, 进行精确的安全分析非常困难。对严肃的应用场合来说, 这种分析是必要的。

本书的最后一个协议是书中第一个协议的一种具体实现。在学习全书之后, 我们现在已经有了很好地实现协议 1.1 的技术准备。这个实现便是著名的 Blum“电话掷币”协议[44]。

### 19.1 Blum“电话掷币”协议

描述在协议 19.1 中的 Blum 远程掷币协议是并行运行的, 它使得互不信任的双方可以协商一个都信赖的  $m$  比特的随机数。如同协议 1.1, 在 Blum 协议中 Alice 掷币, Bob 猜测正反面。

Blum 协议使用了一个大合数  $N = PQ$ , 其中  $P, Q$  是两个大素数, 满足

$$P \equiv Q \equiv 3 \pmod{4}$$

Blum 协议[44]发表后,上述整数便被称为 **Blum 整数**。Blum 整数对密码学用途有很多有用的性质。在 6.7 节我们已经学习了关于 Blum 整数的一些数论事实,其中有些事实对我们在这里分析 Blum 协议的安全性有用。

我们首先对 Blum 远程掷币协议给出一个安全性分析,然后讨论协议的效率。

### 协议 19.1 Blum“电话掷币”协议

( \* 该协议使 Alice 和 Bob 可以协商一个彼此都信赖的  $m$  比特的随机数;如同协议 1.1, Alice 掷币, Bob 猜测 \* )

约定

- 每一方对发往对方的消息都进行数字签名。
  - 每一方如果发现任何验证不一致(含数字签名)就中止运行协议。
1. Bob 生成一个大的 Blum 整数  $N = PQ$  并发送  $N$  给 Alice;
  2. Alice 选取  $m$  个随机数:  $x_1, x_2, \dots, x_m \in_U \mathbb{Z}_N^*$ , 令  $\frac{x_i}{N} (i = 1, 2, \dots, m)$  为掷币结果, 计算  $y_1 \leftarrow x_1^2, y_2 \leftarrow x_2^2, \dots, y_m \leftarrow x_m^2 \pmod{N}$ ; 发送  $y_1, y_2, \dots, y_m$  给 Bob;
  3. Bob 随机选择符号  $b_1, b_2, \dots, b_m \in_U \{1, -1\}$  作为对  $\left(\frac{x_i}{N}\right) (i = 1, 2, \dots, m)$  的符号的猜测; 发送这些符号给 Alice;  
( \* Bob 已经完成了对 Alice 掷币的猜测 \* )
  4. Alice 向 Bob 出示  $x_1, x_2, \dots, x_m$ ;  
( \* Alice 告诉 Bob 正确的猜测 \* )
  5. 对  $i = 1, 2, \dots, m$ ; Bob 验证  $y_i \equiv x_i^2 \pmod{N}$ ; 向 Alice 出示  $P, Q$ ;
  6. Alice 验证  $P \equiv Q \equiv 3 \pmod{4}$ , 对  $P$  和  $Q$  进行素性测试;
  7. 对  $i = 1, 2, \dots, m$ , 双方计算协商的比特值如下:

$$r_i \leftarrow \begin{cases} 1 & \text{如果 Bob 猜测正确, 即 } \left(\frac{x_i}{N}\right) = b_i \\ 0 & \text{其他} \end{cases}$$

## 19.2 安全性分析

在 Blum 远程掷币协议中, Alice 掷币, Bob 猜测正反面。因此在协议的安全性分析中, 我们需要度量双方实施如下两种可能攻击的困难性。

### Alice 欺骗

Alice 能否找到一种方法, 使得她掷币以后能根据自己的意愿出示正反面且让 Bob 相信?

### Bob 不公平猜测的优势

Bob 的猜测优势可以不等于  $1/2$  吗?

我们可以定量地回答这两个问题。首先, Alice 欺骗等价于她分解 Blum 整数  $N$  这一问题; 其次, Bob 的猜测优势正好是  $1/2$ 。现在分别在这里分析。

### 抗 Alice 欺骗的安全性

为了欺骗, Alice 必须找到一对碰撞, 即两个元素  $z_1, z_2 \in \mathbb{Z}_N^*$ , 满足

- $z_1^2 \equiv z_2^2 \pmod{N}$

- $\left(\frac{z_1}{N}\right) \neq \left(\frac{z_2}{N}\right)$

假设 Alice 确实能找到这么一对碰撞。根据定理 6.18 (i), 我们有  $\left(\frac{-1}{N}\right) = 1$ 。这要求  $z_1 \not\equiv \pm z_2 \pmod{N}$ , 即  $0 < z_1 \pm z_2 < N$ 。假定与此相反, 即  $z_1 \equiv -z_2 \pmod{N}$ , 则我们有

$$\left(\frac{z_1}{N}\right) = \left(\frac{-1}{N}\right) \left(\frac{z_2}{N}\right) = \left(\frac{z_2}{N}\right)$$

这与 Alice 的碰撞标准  $\left(\frac{z_1}{N}\right) \neq \left(\frac{z_2}{N}\right)$  相矛盾。

现在根据

$$0 < z_1 + z_2 < N, 0 < |z_1 - z_2| < N$$

和

$$z_1^2 - z_2^2 = (z_1 + z_2)(z_1 - z_2) \equiv 0 \pmod{N}$$

我们得到(例如)

$$0 < \gcd(z_1 + z_2, N) < N$$

也就是说, Alice 已经分解了  $N$ 。

到这里, 我们得出结论, Alice 找到一对碰撞的困难性完全等同于分解  $N$  这样一个著名的困难问题。这里再一次使用了“归约到矛盾”的安全性分析方法。对使用分解的困难性作为量化办法来描述“奇妙函数”性质中的第 2 个“不可能”, 我们是满意的。实际上, 这是一个众所周知的不可能问题, 尤其是考虑到 Alice 还需要实时完成分解工作的情况下更是如此。

因此, 在 Alice 看来, Blum 协议中用来发送掷币承诺的函数确实是一个单向函数, 其可信性基于“种系”问题。

### Bob 的猜测优势

我们现在证明 Bob 的猜测优势正好是  $1/2$ 。

对第  $i$  次掷币, Alice 发送  $y_i \equiv x_i^2 \pmod{N}$  给 Bob。Bob 的工作是在看到  $y_i$  后猜测  $\left(\frac{x_i}{N}\right)$  的符号。根据定理 6.18(iii),  $y_i$  恰有两个平方根的雅可比符号为正、两个平方根的雅可比符号为负。利用算法 6.5, Bob 可以计算出这四个平方根, 但他无法知道 Alice 选择的是哪一个, 因此也就无法确定对应的雅可比符号。对 Bob 来说, 这个函数正好是 2 对 1 映射, 他所能做的就是纯粹的猜测, 其正确概率正好是  $1/2$ 。

这就是我们对“奇妙函数”性质描述中的第 1 个“不可能”的定量度量。这个“不可能”是绝对的!

## 19.3 效率

通过考察协议, 我们可以度量双方计算上的开销如下。

### Alice 的开销

Alice 的主要开销为三个方面开销的和: (i) 计算  $m$  次平方, (ii)  $m$  个雅可比符号, (iii) 进行 2

次素性测试。计算平方和雅可比符号的开销为  $O_B((\log N)^2)$ , 素性测试的开销为  $O_B(\log N)^3$ 。因此, 如果我们令  $m = \log N$ , 则 Alice 的总开销为  $C \cdot (\log N)^3$ , 这里  $C$  是一个很小的常数。这一估计包含了生成和验证数字签名的开销。一般来说, Alice 总的计算开销与当做几次 RSA 加密的开销差不多。

在通信带宽消耗方面, Alice 要发送  $2(\log N)^2$  比特(考虑  $m = \log N$ )。

### Bob 的开销

很清楚, Bob 计算上的开销是 Alice 的开销加上产生一个 RSA 模数的开销。因此, 一般来说, Bob 的计算开销是产生一个 RSA 密钥的开销加上进行几次 RSA 加密的开销。为了表示得正式一些, 我们可以用  $\log N$  替代 Alice 计算上开销中的常数  $C$ , 得到 Bob 计算上开销的表达式:  $O_B((\log N)^4)$ 。

由于 Bob 仅需发送模数、 $m$  个随机比特和该模数的因子, 累计  $3 \log N$ (比特), 因此他的通信开销远远低于 Alice 的开销。

显然, 双方的开销都是适合实际应用的。

## 19.4 本章小结

通过对 Blum“电话掷币”协议的性能及安全性的量化, 我们得到协议的如下特性:

- 可量化的强安全性

从 19.2 节的安全分析可以看到, 电话掷币协议中单向函数的最终实现在可度量意义上是非常强的: 使用“不可能”一词是基于一类“种系”问题——大整数分解的单向性; 另一个“不可能”是绝对意义上的, 即无条件的。

- 实用的效率

从 19.3 节的性能分析中我们还看到, 该协议允许双方以执行若干次普通公钥运算的开销, 协商一个都信赖的  $m$  比特长的随机串, 其中公钥加密系统使用  $m$  作为安全参数。这一效率显然适合实际应用。

- 基于可获得的实用原型

该协议使用普通数字签名方案, 包括计算平方、模一个大整数的雅可比符号和 Monte-Carlo 素性测试。在绝大多数的密码算法库中都有这些标准的算法和运算, 因而可广泛获取。

因此, 根据第 1 章列出的、衡量一个好的密码算法、协议和系统的标准, Blum“电话掷币”协议确实是一个好协议。



## 第 20 章 结 束 语

20 世纪 70 年代中期发生的两件大事标志着进入现代密码学时代——美国数据加密标准的公布和公钥密码学结果的发现。从此,密码学在理论和实用上的重要性就成功地促进着学术研究和商业应用活动的发展。目前,现代密码学已经发展为一个非常广泛的研究领域。随着新的思想和技术不断涌现,该领域的研究也在不断深化。

本书中,我们仅限于学习从现代密码学中精选的很小但很重要的一部分。包括技术、方案、协议和系统,它们或者在信息安全系统设计中起着最基本组成模块的作用(例如,第 7 章至第 10 章的密码学原型和第 11 章的基本认证协议设计),或者已经有最为广泛的应用(例如,第 12 章中现实的认证系统和第 15 章至第 16 章中适于应用的加密和签名方案),或者对构建未来电子商业、商务和服务的“新奇”应用具有潜在的影响(例如,第 13 章基于身份的方案和第 18 章的零知识协议)。

集中研究重点,我们才能够从几个方面系统深入地学习所精选的技术,不仅对应用中正确使用所选择的技术有重要意义,而且对进一步发展信息安全方法也有重要意义。这些方面包括:

- 揭示“教科书式”密码方案和协议的普遍缺陷。
- 将一般安全性概念强化为适于应用的版本。
- 介绍适于应用的密码方案和协议。
- 给出安全性分析的形式化方法和技术。
- 示例一些密码学方案和协议强安全性证据的形式化建立。

此外,我们还学习了现代密码学的基础理论,目的是向读者提供一份入门材料,帮助她/他进一步探索现代密码学这一广阔领域。

## 参考文献

- [1] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. Technical Report DEC SRC Technical Report 125, Digital Equipment Corporation, November 1995.
- [2] M. Abadi and P. Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *Journal of Cryptology*, 15(2):103–127, Spring 2002.
- [3] M. Abadi and M.R. Tuttle. A semantics for a logic of authentication (extended abstract). In *Proceedings of Tenth Annual ACM Symposium on Principles of Distributed Computing*, pages 201–216, August 1991.
- [4] M. Abdalla, M. Bellare, and P. Rogaway. DHAES: an encryption scheme based on the Diffie-Hellman problem. Submission to IEEE P1363: Asymmetric Encryption, 1998. Available at [grouper.ieee.org/groups/1363/P1363a/Encryption.html](http://grouper.ieee.org/groups/1363/P1363a/Encryption.html).
- [5] C. Abrams and A. Drobik. E-business opportunity index — the EU surges ahead. Research Note, Strategic Planning, SPA-10-7786, GartnerGroup RAS Services, 21, July 2000.
- [6] C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). The Internet Engineering Task Force Request For Comments (IETF RFC) 3161, August 2001. Available at [www.ietf.org/rfc/rfc3161.txt](http://www.ietf.org/rfc/rfc3161.txt).
- [7] C. Adams and S. Farrell. Internet X.509 Public Key Infrastructure Certificate Management Protocols. The Internet Engineering Task Force Request For Comments (IETF RFC) 2510, March 1999. Available at [www.ietf.org/rfc/rfc2510.txt](http://www.ietf.org/rfc/rfc2510.txt).
- [8] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Online News, August 2002. [www.cse.iitk.ac.in/users/manindra/primality.ps](http://www.cse.iitk.ac.in/users/manindra/primality.ps).
- [9] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Publishing Company, 1974.
- [10] W. Aiello, S.M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A.D. Keromytis, and O. Reingold. Efficient, DoS-resistant, secure key exchange for Internet Protocols. In B. Christianson et al., editor, *Proceedings of Security Protocols, Lecture Notes in Computer Science 2467*, pages 27–39. Springer-Verlag, 2002.
- [11] W. Aiello, S.M. Bellovin, M. Blaze, R. Canetti, J. Ioannidis, A.D. Keromytis, and O. Reingold. Efficient, DoS-resistant, secure key exchange for Internet Protocols. In *Proceedings of ACM Conference on Computer and Communications Security (ACM-CCS'02)*, pages 48–58. ACM Press, November 2002.

- [12] Alctel. Understanding the IPsec protocol suite. White Papers Archive, March 2000. Available at [www.ind.alctel.com/library/whitepapers/wp\\_IPSec.pdf](http://www.ind.alctel.com/library/whitepapers/wp_IPSec.pdf).
- [13] W. Alexi, B. Chor, O. Goldreich, and C.P. Schnorr. RSA and Rabin functions: certain parts are as hard as the whole. *SIAM Journal of Computing*, 17(2):194–209, April 1988.
- [14] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., 2001.
- [15] R. Anderson, E. Biham, and L. Knudsen. Serpent: A proposal for the advanced encryption standard. AES proposal: National Institute of Standards and Technology (NIST), 1998. Also available at [www.cl.cam.ac.uk/~rja14/serpent.html](http://www.cl.cam.ac.uk/~rja14/serpent.html).
- [16] L. Babai. Talk presented at the 21st Annual Symposium on Foundation of Computer Science. San Juan, Puerto Rico, October 1979.
- [17] R. Baldwin and R. Rivest. The RC5, RC5-CBC, RC5-CBC-Pad, and RC5-CTS algorithms. The Internet Engineering Task Force Request For Comments (IETF RFC) 2040, October 1996. Available at [www.ietf.org/rfc/rfc2040.txt](http://www.ietf.org/rfc/rfc2040.txt).
- [18] N. Barić and B. Pfitzmann. Collision-free accumulations and fail-stop signature schemes without trees. In W. Fumy, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'97, Lecture Notes in Computer Science 1233*, pages 480–494. Springer-Verlag, 1997.
- [19] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key-exchange protocols. In *Proceedings of the 30th Annual Symposium on the Theory of Computing (STOC'98)*, pages 419–428. ACM Press, 1998.
- [20] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. In H. Krawczyk, editor, *Advances in Cryptology — Proceedings of CRYPTO'98, Lecture Notes in Computer Science 1462*, pages 26–45. Springer-Verlag, 1998.
- [21] M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. In G. Brassard, editor, *Advances in Cryptology — Proceedings of CRYPTO'89, Lecture Notes in Computer Science 435*, pages 547–557. Springer-Verlag, 1990.
- [22] M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In B. Preneel, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'00, Lecture Notes in Computer Science 1807*, pages 139–155. Springer-Verlag, 2000.
- [23] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. In *First ACM Conference on Computer and Communications Security*, pages 62–73, New York, 1993. ACM Press.
- [24] M. Bellare and P. Rogaway. Entity authentication and key distribution. In D. Stinson, editor, *Advances in Cryptology — Proceedings of CRYPTO'93, Lecture Notes in Computer Science 773*, pages 232–249. Springer-Verlag, 1994.

- 
- [25] M. Bellare and P. Rogaway. Optimal asymmetric encryption. In A. de Santis, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science 950*, pages 92–111. Springer-Verlag, 1995.
- [26] M. Bellare and P. Rogaway. Provably secure session key distribution — the three party case. In *Proceedings of 27th ACM Symposium on the Theory of Computing*, pages 57–66. ACM Press, 1995.
- [27] M. Bellare and P. Rogaway. The exact security of digital signatures – How to sign with RSA and Rabin. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070*, pages 399–416. Springer-Verlag, 1996.
- [28] S.M. Bellovin. Problem areas for the IP security protocols. In *Proceedings of the Sixth Usenix UNIX Security Symposium*, pages 1–16, July 1996.
- [29] S.M. Bellovin and M. Merritt. Limitations of the Kerberos authentication system. *ACM Computer Communication Review*, 20(5):119–132, 1990.
- [30] S.M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proceedings of the 1992 IEEE Symposium on Research in Security and Privacy*, 1992.
- [31] J.C. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections. In *Proceedings of the 26th Annual Symposium on the Theory of Computing (STOC'94)*, pages 544–553, 1994.
- [32] C. Bennett and G. Brassard. The dawn of a new era for quantum cryptography: the experimental prototype is working! *SIGACT News*, 20:78–82, Fall 1989.
- [33] R. Berger, S. Kannan, and R. Peralta. A framework for the study of cryptographic protocols. In H.C. Williams, editor, *Advances in Cryptology — Proceedings of CRYPTO'85, Lecture Notes in Computer Science 218*, pages 87–103. Springer-Verlag, 1986.
- [34] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4:3–72, 1991.
- [35] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. Systematic design of two-party authentication protocols. In J. Feigenbaum, editor, *Advances in Cryptology — Proceedings of CRYPTO'91, Lecture Notes in Computer Science 576*, Springer-Verlag, pages 44–61, 1992.
- [36] I. Blake, G. Seroussi, and N. Smart. *Elliptic Curves in Cryptography*. Cambridge University Press, 1999. London Mathematical Society Lecture Note Series 265.
- [37] S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In *Proceedings of the sixth IMA International Conference on Cryptography and Coding, Lecture Notes in Computer Science, 1355*, pages 30–45. Springer Verlag, 1997.
- [38] S. Blake-Wilson and A. Menezes. Security proofs for entity authentication and authenticated key transport protocols employing asymmetric techniques. In *Pro-*

*ceedings of 1997 Security Protocols Workshop, Lecture Notes in Computer Science 1361*, pages 137–158. Springer Verlag, 1998.

- [39] S. Blake-Wilson and A. Menezes. Authenticated Diffie-Hellman key agreement protocols. In S. Tavares and H. Meijer, editors, *Proceedings of Selected Areas in Cryptography (SAC'98), Lecture Notes in Computer Science 1556*, pages 339–361. Springer Verlag, 1999.
- [40] M. Blaze. Efficient, DoS-resistant, secure key exchange for Internet protocols (Transcript of Discussion). In B. Christianson et al., editor, *Proceedings of Security Protocols, Lecture Notes in Computer Science 2467*, pages 40–48. Springer-Verlag, 2002.
- [41] M. Blaze, J. Feigenbaum, and J. Lacy. Distributed trust management. In *Proceedings 1996 IEEE Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society Press, May 1996.
- [42] D. Bleichenbacher. Generating ElGamal signature without knowing the secret key. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EURO-CRYPT'96, Lecture Notes in Computer Science 1070*, pages 10–18. Springer-Verlag, 1996.
- [43] L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM Journal of Computing*, 15(2):364–383, May 1986.
- [44] M. Blum. Coin flipping by telephone: A protocol for solving impossible problems. In *Proceedings of the 24th IEEE Computer Conference*, pages 133–137, May 1981.
- [45] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 103–112, 1988.
- [46] M. Blum and S. Goldwasser. An efficient probabilistic public-key encryption scheme which hides all partial information. In G.R. Blakley and D. Chaum, editors, *Advances in Cryptology — Proceedings of CRYPTO'84, Lecture Notes in Computer Science 196*, pages 289–299. Springer-Verlag, 1985.
- [47] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. In *Proceedings of 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 112–117, 1982.
- [48] D. Boneh. The decision Diffie-Hellman problem. In *Proceedings of 3rd Algorithmic Number Theory Symposium, Lecture Notes in Computer Science 1423*, pages 48–63. Springer-Verlag, 1997.
- [49] D. Boneh. Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS*, 46(2):203–213, February 1999.
- [50] D. Boneh. Simplified OAEP for the RSA and Rabin functions. In J. Killian, editor, *Advances in Cryptology — Proceedings of CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 275–291. Springer-Verlag, 2001.

- 
- [51] D. Boneh and G. Durfee. Cryptanalysis of RSA with private key  $d$  less than  $n^{0.292}$ . In J. Stern, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'99, Lecture Notes in Computer Science 1592*, pages 1–11. Springer-Verlag, 1999.
- [52] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. In J. Killian, editor, *Advances in Cryptology — Proceedings of CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 213–229. Springer-Verlag, 2001.
- [53] D. Boneh, A. Joux, and P.Q. Nguyen. Why textbook ElGamal and RSA encryption are insecure (extended abstract). In T. Okamoto, editor, *Advances in Cryptology — Proceedings of ASIACRYPT'00, Lecture Notes in Computer Science 1976*, pages 30–43. Springer-Verlag, 2000.
- [54] A. Bosselaers, H. Dobbertin, and B. Preneel. The new cryptographic hash function RIPEMD-160. *Dr. Dobbs*, 22(1):24–28, January 1997.
- [55] C. Boyd. Hidden assumptions in cryptographic protocols. *IEE Proceedings, Part E*, 137(6):433–436, November 1990.
- [56] C. Boyd and W. Mao. On a limitations of BAN logic. In T. Helleseth, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science 765*, pages 240–247. Springer-Verlag, 1993.
- [57] C. Boyd, W. Mao, and K. Paterson. Deniable authentication for Internet Protocols. In *International Workshop on Security Protocols, Lecture Notes in Computer Science (to appear)*, pages Pre-proceedings: 137–150. Springer-Verlag, April 2003. Sidney Sussex College, Cambridge, England.
- [58] V. Boyko, P. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using Diffie-Hellman. In B. Preneel, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'00, Lecture Notes in Computer Science 1807*, pages 156–171. Springer-Verlag, 2000.
- [59] S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI Technical Report, 1993.
- [60] G. Brassard, D. Chaum, and C. Crépeau. Minimum disclosure proofs of knowledge. *Journal of Computer and System Sciences*, 37(2):156–189, 1988.
- [61] S.C. Brookes, C.A.R. Hoare, and A.W. Roscoe. A theory of communicating sequential processes. *Journal of the Association of Computing Machinery*, 31(7):560–599, 1984.
- [62] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. Technical Report SRC Technical Report 39, Digital Equipment Corporation, February 1989.
- [63] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas Jr., L. O'Connor, M. Peyravian, D. Safford, and N. Zunic. MARS - a candidate cipher for AES. AES proposal: National Institute of Standards and Technology (NIST), 1998. Also available at [www.research.ibm.com/security/mars.html](http://www.research.ibm.com/security/mars.html).



- [64] J. Camenisch and M. Michels. Proving in zero-knowledge that a number is the product of two safe primes. In J. Stern, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'99, Lecture Notes in Computer Science 1592*, pages 106–121. Springer-Verlag, 1999.
- [65] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. In *Proceedings of the 30th Annual Symposium on the Theory of Computing (STOC'98)*, pages 209–218. ACM Press, 1998.
- [66] R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. A new version of [65], October 2002. Available at [xxx.lanl.gov/ps/cs.CR/0010019](http://xxx.lanl.gov/ps/cs.CR/0010019).
- [67] R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In B. Pfitzmann, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'01, Lecture Notes in Computer Science 2045*, pages 453–474. Springer-Verlag, 2001.
- [68] R. Canetti and H. Krawczyk. Security analysis of IKE's signature-based key-exchange protocol. In M. Yung, editor, *Advances in Cryptology — Proceedings of CRYPTO'02, Lecture Notes in Computer Science 2442*, pages 143–161. Springer-Verlag, 2002. Also available at [eprint.iacr.org](http://eprint.iacr.org).
- [69] B. Canvel, A. Hiltgen, S. Vaudenay, and M. Vuagnoux. Password interception in a SSL/TLS channel. To appear in CRYPTO'03, March 2003. Available at [lasecwww.epfl.ch/memo\\_ssl.shtml](http://lasecwww.epfl.ch/memo_ssl.shtml).
- [70] U. Carlsen. Cryptographic protocol flaws: know your enemy. In *Proceedings of The Computer Security Foundations Workshop VII*, pages 192–200. IEEE Computer Society Press, 1994.
- [71] S. Cavallar, B. Dodson, A.K. Lenstra, W. Lioen, P.L. Montgomery, B. Murphy, H. te Riele, K. Aardal, J. Gilchrist, G. Guillerm, P. Leyland, J. Marchand, F. Morain, A. Muffett, C. Putnam, C. Putnam, and P. Zimmermann. Factorization of a 512-bit RSA modulus. In B. Preneel, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'00, Lecture Notes in Computer Science 1807*, pages 1–18. Springer-Verlag, 2000.
- [72] D. Chaum. Demonstrating that a public predicate can be satisfied without revealing any information about how. In A.M. Odlyzko, editor, *Advances in Cryptology — Proceedings of CRYPTO'86, Lecture Notes in Computer Science 263*, pages 195–199. Springer-Verlag, 1987.
- [73] D. Chaum. Zero-knowledge undeniable signatures (extended abstract). In I.B. Damgård, editor, *Advances in Cryptology — Proceedings of CRYPTO'90, Lecture Notes in Computer Science 473*, pages 458–464. Springer-Verlag, 1991.
- [74] D. Chaum and T.P. Pedersen. Wallet databases with observers. In E.F. Brickell, editor, *Advances in Cryptology — Proceedings of CRYPTO'92, Lecture Notes in Computer Science 740*, pages 89–105. Springer-Verlag, 1993.
- [75] D. Chaum and H. van Antwerpen. Undeniable signatures. In G. Brassard, editor, *Advances in Cryptology — Proceedings of CRYPTO'89, Lecture Notes in Computer Science 435*, pages 212–216. Springer-Verlag, 1990.

- [76] B. Chor. *Two Issues in Public Key Cryptography, RSA Bit Security and a New Knapsack Type System*. MIT Press, 1985. An ACM Distinguished Dissertation.
- [77] B. Chor and O. Goldreich. RSA/Rabin least significant bits are  $\frac{1}{2} + \frac{1}{\text{poly}(\log N)}$  secure. In G.T. Blakley and D. Chaum, editors, *Advances in Cryptology — Proceedings of CRYPTO'84, Lecture Notes in Computer Science 196*, pages 303–313. Springer-Verlag, 1985.
- [78] J. Clark and J. Jacob. A survey of authentication protocol literature: version 1.0. Online document, November 1997. Available at [www.cs.york.ac.uk/jac/papers/drareview.ps.gz](http://www.cs.york.ac.uk/jac/papers/drareview.ps.gz).
- [79] C. Cocks. An identity-based public-key cryptosystem. In *Cryptography and Coding: 8th IMA International Conference, Lecture Notes in Computer Science 2260*, pages 360–363. Springer, December 2001.
- [80] H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer, 1996. Graduate Texts in Mathematics 138.
- [81] S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings of 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.
- [82] D. Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. *IBM Journal of Research and Development*, 38:243–250, 1994.
- [83] D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070*, pages 178–189. Springer-Verlag, 1996.
- [84] J.S. Coron, M. Joye, D. Naccache, and P. Paillier. Universal padding schemes for RSA. In M. Yung, editor, *Advances in Cryptology — Proceedings of CRYPTO'02, Lecture Notes in Computer Science 2442*, pages 226–241. Springer-Verlag, 2002.
- [85] R. Cramer and V. Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In H. Krawczyk, editor, *Advances in Cryptology — Proceedings of CRYPTO'98, Lecture Notes in Computer Science 1462*, pages 13–25. Springer-Verlag, 1998.
- [86] R. Cramer and V. Shoup. Signature schemes based on the strong RSA assumption. In *Proceedings of 6th ACM Conference on Computer and Communication Security*. ACM Press, November 1999.
- [87] J. Daemen and V. Rijmen. AES Proposal: Rijndael. AES proposal: National Institute of Standards and Technology (NIST), October 6 1998. Available at [csrc.nist.gov/encryption/aes/](http://csrc.nist.gov/encryption/aes/).
- [88] J. Daemen and V. Rijmen. *The Design of Rijndael: AES — the Advanced Encryption Standard*. Springer-Verlag, 2002. ISBN: 3540425802.
- [89] I. Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In J. Feigenbaum, editor, *Advances in Cryptology — Proceedings of CRYPTO'91, Lecture Notes in Computer Science 576*, pages 445–456. Springer-Verlag, 1992.

- [90] D.W. Davies and W.L. Price. *Security for Computer Networks, An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer (second edition)*. John Wiley & Sons, 1989.
- [91] D. Davis and R. Swick. Workstation services and Kerberos authentication at Project Athena. Technical Memorandum TM-424, MIT Laboratory for Computer Science, February 1990.
- [92] R.A. DeMillo, G.L. Davida, D.P. Dobkin, M.A. Harrison, and R.J. Lipton. *Applied Cryptology, Cryptographic Protocols, and Computer Security Models*, volume 29. Providence: American Mathematical Society, 1983. Proceedings of Symposia in Applied Mathematics.
- [93] R.A. DeMillo and M.J. Merritt. Protocols for data security. *Computer*, 16(2):39–50, February 1983.
- [94] D. Denning. *Cryptography and Data Security*. Addison-Wesley Publishing Company, Inc., 1982.
- [95] D.E. Denning and G.M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):533–536, August 1981.
- [96] T. Dierks and C. Allen. The TLS Protocol, Version 1.0. Request for Comments: 2246, January 1999.
- [97] W. Diffie. The first ten years of public key cryptology. In G.J. Simmons, editor, *Contemporary Cryptology, the Science of Information Integrity*, pages 135–175. IEEE Press, 1992.
- [98] W. Diffie and M. Hellman. Multiuser cryptographic techniques. In *Proceedings of AFIPS 1976 NCC*, pages 109–112. AFIPS Press, Montvale, N.J., 1976.
- [99] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Trans. Info. Theory*, IT-22(6):644–654, 1976.
- [100] W. Diffie, P.C. van Oorschot, and M. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2:107–125, 1992.
- [101] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. In *Proceedings of 23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, 1991. Journal version in *SIAM Journal on Computing*, vol 30, no. 2, 391–437, 2000.
- [102] D. Dolev and A.C. Yao. On the security of public key protocols. In *Proceedings of IEEE 22nd Annual Symposium on Foundations of Computer Science*, pages 350–357, 1981.
- [103] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31(4):469–472, July 1985.
- [104] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. The Internet Engineering Task Force Request For Comments (IETF RFC) 2693, September 1999. Available at [www.ietf.org/rfc/rfc2693.txt](http://www.ietf.org/rfc/rfc2693.txt).
- [105] eMarketer. Security online: Corporate & consumer protection, e-telligence for business. eMarketer Report, February 2003. Available at [www.emarketer.com](http://www.emarketer.com).

- 
- [106] A. Evans Jr., W. Kantrowitz, and E. Weiss. A user authentication scheme not requiring secrecy in the computer. *Communications of the ACM*, 17(8):437–442, 1974.
- [107] U. Feige, A. Fiat, and A. Shamir. Zero-knowledge proofs of identity. *ACM Special Interest Group on Algorithms and Computation Theory (SIGACT)*, pages 210–217, 1987.
- [108] H. Feistel. Cryptography and computer privacy. *Sci. Am.*, 228(5):15–23, May 1974.
- [109] N. Ferguson and B. Schneier. A cryptographic evaluation of IPsec. Counterpane Labs, 2000. Available at [www.counterpane.com/ipsec.pdf](http://www.counterpane.com/ipsec.pdf).
- [110] A. Fiat and A. Shamir. How to prove yourself: practical solutions of identification and signature problems. In A.M. Odlyzko, editor, *Advances in Cryptology — Proceedings of CRYPTO'86, Lecture Notes in Computer Science 263*, pages 186–194. Springer-Verlag, 1987.
- [111] Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics & Chip Design*. O'Reilly & Associates, May 1998. ISBN 1-56592-520-3.
- [112] A.O. Freier, P. Karlton, and P.C. Kocher. The SSL Protocol, Version 3.0. INTERNET-DRAFT, draft-freier-ssl-version3-02.txt, November 1996.
- [113] E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B.S. Kaliski Jr., editor, *Advances in Cryptology — Proceedings of CRYPTO'97, Lecture Notes in Computer Science 1294*, pages 16–30. Springer-Verlag, 1997.
- [114] E. Fujisaki and T. Okamoto. How to enhance the security of public-key encryption at minimum cost. In H. Imai and Y. Zheng, editors, *Public Key Cryptography — Proceedings of PKC'99, Lecture Notes in Computer Science 1560*, pages 53–68. Springer-Verlag, 1999.
- [115] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, *Advances in Cryptology — Proceedings of CRYPTO'99, Lecture Notes in Computer Science 1666*, pages 537–554. Springer-Verlag, 1999.
- [116] E. Fujisaki, T. Okamoto, D. Pointcheval, and J. Stern. RSA-OAEP Is secure under the RSA assumption. In J. Killian, editor, *Advances in Cryptology — Proceedings of CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 260–274. Springer-Verlag, 2001.
- [117] K. Gaarder and E. Sneekenes. Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol. *Journal of Cryptology*, 3(2):81–98, 1991.
- [118] S. Galbraith. Supersingular curves in cryptography. In C. Boyd, editor, *Advances in Cryptology — Proceedings of ASIACRYPT'01, Lecture Notes in Computer Science 2248*, pages 495–513. Springer-Verlag, 2001.

- [119] S.D. Galbraith, W. Mao, and K.G. Paterson. A cautionary note regarding cryptographic protocols based on composite integers. Technical Report HPL-2001-284, HP Laboratories, Bristol, November 2001.
- [120] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.
- [121] C.F. Gauss. *Disquisitiones Arithmeticae*. Translated by A. Arthur and S.J. Clark, 1996, Yale University Press, New Haven, 1801.
- [122] R. Gennaro, D. Miccianicio, and T. Rabin. An efficient non-interactive statistical zero-knowledge proof system for quasi-safe prime products. In *5th ACM Conference on Computer and Communications Security, Fairfax, Virginia*, 1998.
- [123] M. Girault. An identity-based identification scheme based on discrete logarithms modulo a composite number. In I.B. Damgård, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'90, Lecture Notes in Computer Science 473*, pages 481–486. Springer-Verlag, 1991.
- [124] M. Girault. Self-certified public keys. In D.W. Davies, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science 547*, pages 490–497. Springer-Verlag, 1991.
- [125] I. Goldberg and D. Wagner. Randomness and the Netscape browser, how secure is the World Wide Web? *Dr. Dobbs' Journal*, pages 66–70, January 1996.
- [126] O. Goldreich, S. Micali, and A. Wigderson. How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design (extended abstract). In A.M. Odlyzko, editor, *Advances in Cryptology — Proceedings of CRYPTO'86, Lecture Notes in Computer Science 263*, pages 171–185. Springer-Verlag, 1987.
- [127] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28:270–299, 1984.
- [128] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proceedings of 17th Ann. ACM Symp. on Theory of Computing*, pages 291–304, 1985. A journal version under the same title appears in: *SIAM Journal of Computing* vol. 18, pp. 186–208, 1989.
- [129] S. Goldwasser, S. Micali, and R.L. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM Journal of Computing*, 17(2):281–308, 1988.
- [130] S. Goldwasser, S. Micali, and P. Tong. Why and how to establish a private code on a public network. In *Proceedings of 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 134–144, 1982.
- [131] D. Gollmann. *Computer Security*. John Wiley & Sons, Inc., 1999. ISBN: 0-471-97884-2.
- [132] D. Gollmann. Authentication — myths and misconceptions. *Progress in Computer Science and Applied Logic*, 20:203–225, 2001. Birkhäuser Verlag Basel/Switzerland.

- 
- [133] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*, pages 234–248. IEEE Computer Society Press, 1990.
- [134] F.T. Grampp and R.H. Morris. Unix operating system security. *AT&T Bell Laboratories Technical Journal*, 63(8):1649–1672, October 1984.
- [135] C.G. Günther. An identity-based key-exchange protocol. In J.-J. Quisquater and J. Vanderwalle, editors, *Advances in Cryptology — Proceedings of EURO-CRYPT'89, Lecture Notes in Computer Science 434*, pages 29–37. Springer-Verlag, 1990.
- [136] N.M. Haller. The S/KEY one-time password system. In *Proceedings of the Symposium on Network and Distributed System Security*, pages 151–157, 1994.
- [137] D. Harkins and D. Carrel. The Internet key exchange protocol (IKE). The Internet Engineering Task Force Request For Comments (IETF RFC) 2409, November 1998. Available at [www.ietf.org/rfc/rfc2409.txt](http://www.ietf.org/rfc/rfc2409.txt).
- [138] K.E.B. Hickman. The SSL Protocol. Online document, February 1995. Available at [www.netscape.com/eng/security/SSL\\_2.html](http://www.netscape.com/eng/security/SSL_2.html).
- [139] C.A.R. Hoare. Communicating sequential processes. *Communications of the ACM*, 21(8), 1978.
- [140] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, 1985. Series in Computer Science.
- [141] P. Hoffman. Features of proposed successors to IKE. INTERNET-DRAFT, draft-ietf-ipsec-soi-features-01.txt, May 2002.
- [142] R. Housley and P. Hoffman. Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP. The Internet Engineering Task Force Request For Comments (IETF RFC) 2585, August 2001. Available at [www.ietf.org/rfc/rfc2585.txt](http://www.ietf.org/rfc/rfc2585.txt).
- [143] D. Hühnlein, M. Jakobsson, and D. Weber. Towards practical non-interactive public key cryptosystems using non-maximal imaginary quadratic orders. In *Proceedings of Selected Areas of Cryptography — SAC 2000, Lecture Notes in Computer Science 2012*, pages 275–287. Springer-Verlag, 2000.
- [144] ISO/IEC. Information Processing — Modes of operation for an  $n$ -bit block cipher algorithm. International Organization for Standardization and International Electro-technical Commission, 1991. 10116.
- [145] ISO/IEC. Information Technology — Security Techniques — summary of voting on letter ballot No.6, Document SC27 N277, CD 9798-3.3 “Entity Authentication Mechanisms” — Part 3: Entity authentication mechanisms using a public key algorithm. International Organization for Standardization and International Electro-technical Commission, October 1991. ISO/IEC JTC 1/SC27 N313.
- [146] ISO/IEC. Information Technology — Security Techniques — Entity Authentication Mechanisms — Part 2: Entity authentication using symmetric techniques. International Organization for Standardization and International Electro-technical Commission, 1992. ISO/IEC JTC 1/SC 27 N489 CD 9798-2, 1992-06-09.



- [147] ISO/IEC. Information Technology — Security Techniques — Entity Authentication Mechanisms — Part 2: Entity authentication using symmetric techniques. International Organization for Standardization and International Electro-technical Commission, 1993. ISO/IEC JTC 1/SC 27 N739 DIS 9798-2, 1993-08-13.
- [148] ISO/IEC. Information Technology — Security Techniques — Entity Authentication — Part 1: General. International Organization for Standardization and International Electro-technical Commission, 1996. ISO/IEC JTC 1/SC 27 DIS 9798-1: 1996 (E).
- [149] ISO/IEC. Information Technology — Security Techniques — Entity Authentication — Part 2: Mechanisms using symmetric encipherment algorithms. International Organization for Standardization and International Electro-technical Commission, December 1998. ISO/IEC JTC 1/SC 27 N2145 FDIS 9798-2.
- [150] ISO/IEC. Information Technology — Security Techniques — Entity Authentication — Part 3: Mechanisms using digital signature techniques. International Organization for Standardization and International Electro-technical Commission, October 1998. BS ISO/IEC 9798-3.
- [151] ISO/IEC. Information Technology — Security Techniques — Entity Authentication — Part 4: Mechanisms using a cryptographic check function. International Organization for Standardization and International Electro-technical Commission, April 1999. ISO/IEC JTC 1/SC 27 N2289 FDIS 9798-4.
- [152] ISO/IEC. Information technology — Security techniques — Digital signature schemes giving message recovery — Part 3: Discrete logarithm based mechanisms. International Organization for Standardization and International Electro-technical Commission, April 2000. ISO/IEC JTC 1/SC 27 9796-3.
- [153] ISO/IEC. Information Technology — Security Techniques — Hash Functions — Part 3: Dedicated hash-functions. International Organization for Standardization and International Electro-technical Commission, November 2001. ISO/IEC JTC1, SC27, WG2, Document 1st CD 10118-3.
- [154] ITU-T. Rec. X.509 (revised) the Directory — Authentication Framework, 1993. International Telecommunication Union, Geneva, Switzerland (equivalent to ISO/IEC 9594-8:1995.).
- [155] M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070*, pages 143–154. Springer-Verlag, 1996.
- [156] A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *Algorithmic Number Theory, IV-th Symposium (ANTS IV), Lecture Notes in Computer Science 1838*, pages 385–394. Springer-Verlag, 2000.
- [157] A. Joux and K. Nguyen. Separating decision Diffie-Hellman from Diffie-Hellman in cryptographic groups. *Cryptology ePrint Archive*, 2001/003, 2001. Available at <http://eprint.iacr.org/>.
- [158] R. Kailar. Accountability in electronic commerce protocols. *IEEE Transactions on Software Engineering*, 22(5):313–328, May 1996.

- 
- [159] C. Kaufman. Comparison of IKEv2, JFK, and SOI requirements. The Internet Engineering Task Force: online document, April 2002. Available at [www.ietf.org/proceedings/02mar/slides/ipsec-1/](http://www.ietf.org/proceedings/02mar/slides/ipsec-1/).
  - [160] C. Kaufman. Internet Key Exchange (IKEv2) Protocol. The Internet Engineering Task Force: INTERNET-DRAFT, draft-ietf-ipsec-ikev2-03.txt, October 2002. Available at [www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-03.txt](http://www.ietf.org/internet-drafts/draft-ietf-ipsec-ikev2-03.txt).
  - [161] C. Kaufman, R. Perlman, and M. Speciner. *Network Security: Private Communication in a Public World, Second Edition*. Prentice-Hall PTR, 2002.
  - [162] R. Kemmerer, C. Meadows, and J. Millen. Three systems for cryptographic protocol analysis. *Journal of Cryptology*, 7(2):79–130, 1994.
  - [163] S. Kent and R. Atkinson. IP Authentication Header. The Internet Engineering Task Force Request For Comments (IETF RFC) 2402, November 1998. Available at [www.ietf.org/rfc/rfc2402.txt](http://www.ietf.org/rfc/rfc2402.txt).
  - [164] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). The Internet Engineering Task Force Request For Comments (IETF RFC) 2406, November 1998. Available at [www.ietf.org/rfc/rfc2406.txt](http://www.ietf.org/rfc/rfc2406.txt).
  - [165] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. The Internet Engineering Task Force Request For Comments (IETF RFC) 2401, November 1998. Available at [www.ietf.org/rfc/rfc2401.txt](http://www.ietf.org/rfc/rfc2401.txt).
  - [166] J. Klensin. Simple mail transfer protocol. The Internet Engineering Task Force Request For Comments (IETF RFC) 2821, April 2001. Available at [www.ietf.org/rfc/rfc2821.txt](http://www.ietf.org/rfc/rfc2821.txt).
  - [167] L.R. Knudsen. *Block Ciphers — Analysis, Design and Applications*. Århus University, 1994.
  - [168] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(5):203–209, 1987.
  - [169] P.C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In N. Koblitz, editor, *Advances in Cryptology — Proceedings of CRYPTO'96, Lecture Notes in Computer Science 1109*, pages 104–113. Springer-Verlag, 1996.
  - [170] J. Kohl and C. Neuman. The Kerberos network authentication service (v5). The Internet Engineering Task Force Request For Comments (IETF RFC) 1510, September 1993. Available at [www.ietf.org/rfc/rfc1510.txt](http://www.ietf.org/rfc/rfc1510.txt).
  - [171] L.M. Kohnfelder. *Towards a Practical Public-key Cryptosystem*. MIT B.S. Thesis, MIT Department of Electrical Engineering, May 1978.
  - [172] E. Kranakis. *Primality and Cryptography*. John Wiley & Sons, 1986. Wiley-Teubner Series in Computer Science.
  - [173] H. Krawczyk. SIGMA: the 'SIGn-and-MAC' approach to authenticated Diffie-Hellman protocols. Online document, 1996. Available at [www.ee.technion.ac.il/~hugo/sigma.html](http://www.ee.technion.ac.il/~hugo/sigma.html).
  - [174] H. Krawczyk. SKEME: a versatile secure key exchange mechanism for Internet.

- [175] L. Lamport. Constructing digital signatures from a one way function. *SIR International*, October 1979. Available at [www.csl.sri.com/papers/676/](http://www.csl.sri.com/papers/676/).
- [176] L. Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, 1981.
- [177] A. Lenstra and E. Verheul. The XTR public key system. In M. Bellare, editor, *Advances in Cryptology — Proceedings of CRYPTO'00, Lecture Notes in Computer Science 1880*, pages 1–19. Springer-Verlag, 2000.
- [178] W.J. LeVeque. *Fundamentals of Number Theory*. Dover Publications, Inc., 1977.
- [179] R. Lidl and H. Niederreiter. *Finite Fields*. Cambridge University Press, 1997. *Encyclopedia of Mathematics and its Applications* 20.
- [180] R.J. Lipton. How to cheat at mental poker. Technical Report, Comp. Sci., Dept. Univ. of Calif., Berkeley, Calif., August 1979. (This is an internal technical report; a simple description of the attack is available in page 174 of [92]).
- [181] G. Lowe. Some new attacks upon security protocols. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pages 162–169. IEEE Computer Society Press, June 1994.
- [182] G. Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Information Processing Letters*, 56(3):131–133, 1995.
- [183] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using CSP and FDR. In *Proceedings of TACAS, Lecture Notes in Computer Science 1055*, pages 147–166. Springer-Verlag, 1996.
- [184] J. Malone-Lee and W. Mao. Two birds one stone: Signcryption using RSA. In M. Joye, editor, *Topics in Cryptology — the Cryptographers' Track, Proceedings of the RSA Conference 2003 (CT-RSA 2003), Lecture Notes in Computer Science 2612*, pages 210–224. Springer-Verlag, April 2003.
- [185] W. Mao. An augmentation of BAN-like logics. In *Proceedings of Computer Security Foundations Workshop VIII*, pages 44–56. IEEE Computer Society Press, June 1995.
- [186] W. Mao and C. Boyd. On the use of encryption in cryptographic protocols. In P.G. Farrell, editor, *Codes and Cyphers — Proceedings of 4th IMA Conference on Cryptography and Coding*, pages 251–262, December 1993. The Institute of Mathematics and Its Applications, 1995.
- [187] W. Mao and C. Boyd. On the use of encryption in cryptographic protocols, February 1994. Distributed by International Organization for Standardization (ISO) and International Electro-technical Commission (IEC) JTC1, SC27, WG2, Document N262: “Papers on authentication and key management protocols based on symmetric techniques.” This ISO document distributes the paper published in [186].
- [188] W. Mao and C. Boyd. Methodical use of cryptographic transformations in authentication protocols. *IEE Proceedings, Comput. Digit. Tech.*, 142(4):272–278, July 1995.

- 
- [189] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet security association and key management protocol (ISAKMP), version 10. INTERNET-DRAFT: draft-ietf-ipsec-isakmp-10.txt, November 1998. Also available at [www.ietf.org/rfc/rfc2408.txt](http://www.ietf.org/rfc/rfc2408.txt).
- [190] U. Maurer. Protocols for secret key agreement by public discussion based on common information. In E.F. Brickell, editor, *Advances in Cryptology — Proceedings of CRYPTO'92, Lecture Notes in Computer Science 740*, pages 461–470. Springer-Verlag, 1993.
- [191] U. Maurer. Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory*, IT-39:733–742, 1993.
- [192] U. Maurer and S. Wolf. The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms. *SIAM Journal of Computing*, 28(5):1689–1721, 1999.
- [193] U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In D.W. Davies, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science 547*, pages 498–507. Springer-Verlag, 1991.
- [194] C. Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1(1):5–53, 1992.
- [195] C. Meadows. Analyzing the Needham-Schroeder public key protocol: a comparison of two approaches. In E. Bertino et al, editor, *Proceedings of Computer Security, ESORICS'96, Lecture Notes in Computer Science 1146*, pages 351–364. Springer-Verlag, February 1996.
- [196] C. Meadows. The NRL Protocol Analyzer: an overview. *Journal of Logic Programming*, 26(2):113–131, February 1996.
- [197] C. Meadows. Analysis of the internet key exchange protocol using the NRL Protocol Analyzer. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 216–231. IEEE Computer Society Press, May 1999.
- [198] C. Meadows and P. Syverson. A formal specification of requirements for payment transactions in the SET protocol. In R. Hirschfeld, editor, *Proceedings of Financial Cryptography (FC'98), Lecture Notes in Computer Science 1465*, pages 122–140. Springer-Verlag, February 1998.
- [199] A.J. Menezes, T. Okamoto, and S.A. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Trans. Info. Theory*, 39:1636–1646, 1983.
- [200] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
- [201] R.C. Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21:294–299, 1978.
- [202] R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. on Info. Theory*, 24:525–530, 1978.
- [203] S. Micali and R.L. Rivest. Micropayments revisited. In B. Preneel, editor, *Topics in Cryptology — the Cryptographers' Track, Proceedings of the RSA Conference 2002 (CT-RSA 2002), Lecture Notes in Computer Science 2271*, pages 149–163. Springer-Verlag, 2002.

- [204] S.P. Miller, C. Neuman, J.I. Schiller, and J.H. Saltzer. Kerberos authentication and authorization system. Project Athena Technical Plan Section E.2.1, 1987.
- [205] V. Miller. Use of elliptic curves in cryptography. In H.C. Williams, editor, *Advances in Cryptology — Proceedings of CRYPTO'85, Lecture Notes in Computer Science 218*, pages 417–426. Springer-Verlag, 1986.
- [206] J.H. Moore. Protocol failures in cryptosystems. *Proceedings of the IEEE*, 76(5):594–601, 1988.
- [207] J.H. Moore. Protocol failures in cryptosystems. In G.J. Simmons, editor, *Contemporary Cryptology, the Science of Information Integrity*, pages 541–558. IEEE Press, 1992.
- [208] R. Morris and K. Thompson. Password security: a case history. *Communications of the ACM*, 22(5):594–597, 1979.
- [209] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. The Internet Engineering Task Force Request For Comments (IETF RFC) 2560, June 1999. Available at [www.ietf.org/rfc/rfc2560.txt](http://www.ietf.org/rfc/rfc2560.txt).
- [210] M. Myers, X. Liu, J. Schaad, and J. Weinstein. Certificate Management Messages over CMS. The Internet Engineering Task Force Request For Comments (IETF RFC) 2797, April 2000. Available at [www.ietf.org/rfc/rfc2797.txt](http://www.ietf.org/rfc/rfc2797.txt).
- [211] M. Naor and O. Reingold. Number theoretic constructions of efficient pseudo-random functions. In *Proceedings of FOCS'97*, pages 458–467, 1997.
- [212] M. Naor and M. Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of 22nd ACM Symposium of Theory of Computing*, pages 427–437, 1990.
- [213] NBS. Data Encryption Standard. U.S. Department of Commerce, FIPS Publication 46, Washington, D.C., January 1977. National Bureau of Standards.
- [214] R. Needham and M. Schroeder. Authentication revisited. *Operating Systems Review*, 21:7, 1987.
- [215] R.M. Needham and M.D. Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12):993–999, 1978.
- [216] B.C. Neuman and S.G. Stubblebine. A note on the use of timestamps as nonces. *ACM Operating Systems Review*, 27(2):10–14, April 1993.
- [217] NIST. A Proposed Federal Information Processing Standard for Digital Signature Standard (DSS). Federal Register Announcement August 30, 1991. National Institute of Standards and Technology.
- [218] NIST. Digital Signature Standard. Federal Information Processing Standards Publication 186, 1994. U.S. Department of Commerce/N.I.S.T.
- [219] NIST. Secure Hash Standard. Federal Information Processing Standards Publication (FIPS PUB) 180-1, April 1995. U.S. Department of Commerce/N.I.S.T.

- 
- [220] NIST. Recommendation for block cipher modes of operation. NIST Special Publication 800-38A 2001 Edition, December 2001. U.S. Department of Commerce/N.I.S.T.
  - [221] NIST. Specification for the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication (FIPS PUB) 197, November 2001. U.S. Department of Commerce/N.I.S.T.
  - [222] K. Nyberg and R. Rueppel. A new signature scheme based on the DSA giving message recovery. In *1st ACM Conference on Computer and Communications Security*, pages 58–61. ACM Press, 1993.
  - [223] A.M. Odlyzko. Discrete logarithms: the past and the future. *Designs, Codes and Cryptography*, 19:129–154, 2000.
  - [224] K. Ohta and T. Okamoto. On concrete security treatment of signatures derived from identification. In H. Krawczyk, editor, *Advances in Cryptology — Proceedings of CRYPTO'98, Lecture Notes in Computer Science 1462*, pages 345–370. Springer-Verlag, 1998.
  - [225] T. Okamoto and D. Pointcheval. REACT: rapid enhanced-security asymmetric cryptosystem transform. In D. Naccache, editor, *Topics in Cryptography, Cryptographers' Track, RSA Conference 2001 — Proceedings of CT-RSA'00, Lecture Notes in Computer Science 2020*, pages 159–175. Springer-Verlag, 2001.
  - [226] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. In K. Nyberg, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'98, Lecture Notes in Computer Science 1403*, pages 308–318. Springer-Verlag, 1998.
  - [227] H. Orman. The Oakley key determination protocol, version 2. draft-ietf-ipsec-oakley-02.txt, 1996.
  - [228] D. Otway and O. Rees. Efficient and timely mutual authentication. *Operating Systems Review*, 21(1):8–10, 1987.
  - [229] Oxford. *Oxford Reference, Dictionary of Computing, Third Edition*. Oxford University Press, 1991.
  - [230] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'99, Lecture Notes in Computer Science 1592*, pages 223–238. Springer-Verlag, 1999.
  - [231] J. Patarin and L. Goubin. Trapdoor one-way permutations and multivariate polynomials. In Y. Han, T. Okamoto, and S. Qing, editors, *Information and Communications Security — Proceedings of ICICS'97, Lecture Notes in Computer Science 1334*, pages 356–368. Springer-Verlag, 1997.
  - [232] PKCS. Public Key Cryptography Standards, PKCS#1 v2.1. RSA Cryptography Standard, Draft 2, 2001. Available at [www.rsasecurity.com/rsalabs/pkcs/](http://www.rsasecurity.com/rsalabs/pkcs/).
  - [233] S.C. Pohlig and M.E. Hellman. An improved algorithm for computing logarithms over  $GF(p)$  and its cryptographic significance. *IEEE Transactions on Information Theory*, 24:106–110, 1978.



- [234] D. Pointcheval. HD-RSA: hybrid dependent RSA, a new public-key encryption scheme. Submission to IEEE P1363: Asymmetric Encryption, 1999. Available at [grouper.ieee.org/groups/1363/P1363a/Encryption.html](http://grouper.ieee.org/groups/1363/P1363a/Encryption.html).
- [235] D. Pointcheval. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'99, Lecture Notes in Computer Science 1592*, pages 239–254. Springer-Verlag, 1999.
- [236] D. Pointcheval. Chosen-ciphertext security for any one-way cryptosystem. In H. Imai and Y. Zheng, editors, *Public Key Cryptography — Proceedings of PKC'00, Lecture Notes in Computer Science 1751*, pages 129–146. Springer-Verlag, 2000.
- [237] D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070*, pages 387–398. Springer-Verlag, 1996.
- [238] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 13(3):361–396, 2000.
- [239] J.M. Pollard. Theorems on factorization and primality testing. *Proceedings of the Cambridge Philosophical Society*, 76:521–528, 1974.
- [240] J.M. Pollard. Monte Carlo method for index computation (mod  $p$ ). *Mathematics of Computation*, 32(143):918–924, 1978.
- [241] M. Rabin. Transaction protection by beacons. Technical Report Tech.Rep. 29-81, Aiken Computation Lab., Harvard University, Cambridge, MA, 1981.
- [242] M.O. Rabin. Digitized signatures and public-key functions as intractible as factorization. Technical Report LCS/TR-212, MIT Laboratory for Computer Science, 1979.
- [243] C. Rackoff and D. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In J. Feigenbaum, editor, *Advances in Cryptology — Proceedings of CRYPTO'91, Lecture Notes in Computer Science 576*, pages 433–444. Springer-Verlag, 1992.
- [244] R. Rivest and A. Shamir. PayWord and MicroMint: two simple micropayment schemes. *CryptoBytes, RSA Laboratories*, 2(1):7–11, Spring 1996.
- [245] R.L. Rivest. The MD5 message-digest algorithm. Internet Request for Comments 1321, April 1992.
- [246] R.L. Rivest. S-expressions. INTERNET-DRAFT, May 1997. Available at [theory.lcs.mit.edu/~rivest/sexp.txt](http://theory.lcs.mit.edu/~rivest/sexp.txt).
- [247] R.L. Rivest and B. Lampson. SDSI - A simple distributed security infrastructure. Invited Speech at CRYPTO'96, August 1996. Available at [theory.lcs.mit.edu/~cis/sdsi.html](http://theory.lcs.mit.edu/~cis/sdsi.html).
- [248] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

- 
- [249] R. Sidney R.L. Rivest, M.J.B. Robshaw and Y.L. Yin. The RC6 Block Cipher, v1.1. AES proposal: National Institute of Standards and Technology (NIST), 1998. Available at [www.rsa.com/rsalabs/aes/](http://www.rsa.com/rsalabs/aes/).
- [250] A.W. Roscoe. Model checking CSP. In A.W. Roscoe, editor, *A Classical Mind: Essays in honour of C.A.R. Hoare*. Prentice-Hall, 1994.
- [251] A.W. Roscoe. Modelling and verifying key-exchange protocols using CSP and FDR. In *Proceedings of Computer Security Foundations Workshop VIII*, pages 98–107. IEEE Computer Society Press, June 1995.
- [252] P. Ryan and S. Schneider. *The Modelling and Analysis of Security Protocols: the CSP Approach*. Addison-Wesley, 2001.
- [253] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. In *Proceedings of the 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, January 2000*.
- [254] T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comm. Math. Univ. Sancti. Pauli*, 47:81–92, Spring 1998.
- [255] S. Schneider. Security properties and CSP. In *Proceedings of the 1996 IEEE Symposium in Security and Privacy*, pages 174–187. IEEE Computer Society Press, 1996.
- [256] B. Schneier. *Secrets and Lies*. John Wiley & Sons, 2001.
- [257] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. Twofish: a 128-bit block cipher, AES proposal. AES proposal: National Institute of Standards and Technology (NIST), 1998. Available at [www.counterpane.com/twofish.html](http://www.counterpane.com/twofish.html).
- [258] C.P. Schnorr. Efficient identification and signature for smart cards. In G. Brassard, editor, *Advances in Cryptology — Proceedings of CRYPTO'89, Lecture Notes in Computer Science 435*, pages 239–252. Springer-Verlag, 1990.
- [259] C.P. Schnorr. Efficient signature generation for smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
- [260] L.A. Semaev. Evaluation of discrete logarithms in a group of  $p$ -torsion points of an elliptic curve in characteristic  $p$ . *Math. Comp.*, 67(221):353–356, 1998.
- [261] SET. Secure Electronic Transaction Specification, Version 1.0. Online document, May 1997. Available at [www.setco.org/](http://www.setco.org/).
- [262] A. Shamir. Identity-based cryptosystems and signature schemes. In G.T. Blakeley and D. Chaum, editors, *Advances in Cryptology — Proceedings of CRYPTO'84, Lecture Notes in Computer Science 196*, pages 48–53. Springer-Verlag, 1985.
- [263] A. Shamir, R. Rivest, and L. Adleman. Mental poker. In D. Klarner, editor, *The Mathematical Gardner*, pages 37–43, Boston, Mass, 1980. Prindle, Weber & Schmidt.

- [264] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27(3):379–423, July 1948.
- [265] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, 27:623–656, October 1948. Continued from July 1948 issue (i.e., [264]).
- [266] C.E. Shannon. Communications theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, October 1949.
- [267] C.E. Shannon. Predilection and entropy of printed English. *Bell Systems Technical Journal*, 30:50–64, January 1951.
- [268] R. Shirey. Internet Security Glossary. The Internet Engineering Task Force Request For Comments (IETF RFC) 2828, May 2000. Available at [www.ietf.org/rfc/rfc2828.txt](http://www.ietf.org/rfc/rfc2828.txt).
- [269] P.W. Shor. Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computing*, 26:1484–1509, 1997.
- [270] P.W. Shor. Why haven't more quantum algorithms been found? *Journal of the ACM*, 50(1):87–90, January 2003.
- [271] V. Shoup. Using hash functions as a hedge against chosen ciphertext attack. In B. Preneel, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'00, Lecture Notes in Computer Science 1807*, pages 275–288. Springer-Verlag, 2000.
- [272] V. Shoup. OAEP reconsidered. In J. Killian, editor, *Advances in Cryptology — Proceedings of CRYPTO'01, Lecture Notes in Computer Science 2139*, pages 239–259. Springer-Verlag, 2001.
- [273] V. Shoup. A proposal for an ISO standard for public key encryption (version 2.1). Distributed by International Organization for Standardization (ISO) and International Electro-technical Commission (IEC) JTC1, SC27, WG2, December 2001. An earlier version appeared in ISO/IEC JTC 1/SC 27 N2765 “Editor's contribution on public key encryption” (February 2001).
- [274] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Springer-Verlag, 1986. Graduate Texts in Mathematics.
- [275] R.D. Silverman. Fast generation of random, strong RSA primes. *CryptoBytes*, 3(1):9–13, 1997.
- [276] G.J. Simmons. How to (selectively) broadcast a secret. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 108–113. IEEE Computer Society Press, 1985.
- [277] G.J. Simmons. A survey of information authentication. In G.J. Simmons, editor, *Contemporary Cryptology, the Science of Information Integrity*, pages 379–419. IEEE Press, 1992.
- [278] D. Simon. On the power of quantum computation. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 116–123, 1994.

- [279] S. Singh. *The Code Book*. Fourth Estate, 1999.
- [280] N.P. Smart. The discrete logarithm problem on elliptic curves of trace one. *Journal of Cryptology*, 12:193–196, 1999.
- [281] M.E. Smid and D.K. Branstad. The Data Encryption Standard, past and future. In G.J. Simmons, editor, *Contemporary Cryptology, the Science of Information Integrity*, pages 43–46. IEEE Press, 1992.
- [282] D. Soldera. SEG - a provably secure variant of El-Gamal. Technical Report HPL-2001-149, Hewlett-Packard Laboratories, Bristol, June 2001.
- [283] D. Soldera, J. Seberry, and C. Qu. The analysis of Zheng-Seberry scheme. In L. M. Batten and J. Seberry, editors, *7th Australian Conference in Information Security and Privacy — Proceedings of ACISP'02, Lecture Notes in Computer Science 2384*, pages 159–168. Springer-Verlag, 2002.
- [284] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM Journal of Computing*, 6(1):84–85, March 1977.
- [285] M. Stadler. Publicly verifiable secret sharing. In U. Maurer, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'96, Lecture Notes in Computer Science 1070*, pages 190–199. Springer-Verlag, 1996.
- [286] D.R. Stinson. *Cryptography: Theory and Practice*. CRC Press, Inc., 1995.
- [287] P. Syverson. On key distribution protocols for repeated authentication. *ACM Operating Systems Review*, 27(4):24–30, October 1993.
- [288] P. Syverson and P.C. van Oorschot. On unifying some cryptographic protocol logics. In *Proceedings of 1994 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1994.
- [289] H. Tanaka. A realization scheme for the identity-based cryptosystem. In C. Pomerance, editor, *Advances in Cryptology — Proceedings of CRYPTO'87, Lecture Notes in Computer Science 293*, pages 340–349. Springer-Verlag, 1988.
- [290] G. Trudik. Message authentication with one-way functions. *Computer Communication Review*, 22:29–38, 1992.
- [291] S. Tsuji and T. Itoh. An ID-based cryptosystem based on the discrete logarithm problem. *IEEE Journal on Selected Areas in Communication*, 7(4):467–473, 1989.
- [292] W. Tuchman. Hellman presents no shortcut solutions to the DES. *IEEE Spectrum*, 16(7):40–41, 1979.
- [293] G. van de Graaf and R. Peralta. A simple and secure way to show the validity of your public key. In C. Pomerance, editor, *Advances in Cryptology — Proceedings of CRYPTO'87, Lecture Notes in Computer Science 293*, pages 128–134. Springer-Verlag, 1988.
- [294] P.C. van Oorschot. Extending cryptographic logics of belief to key agreement protocols (extended abstract). In *Proceedings of the First ACM Conference on Computer and Communications Security*, pages 232–243, 1993.

- [295] V. Varadharajan, P. Allen, and S. Black. An analysis of the proxy problem in distributed systems. In *Proceedings of the 1991 IEEE Symposium on Security and Privacy*, pages 255–275, 1991.
- [296] S. Vaudenay. Security flaws induced by CBC padding – Applications to SSL, IPSEC, WTLS . . . . In L.R. Knudsen, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'02, Lecture Notes in Computer Science 2332*, pages 534–545. Springer-Verlag, 2002.
- [297] U. Vazirani and V. Vazirani. Efficient and secure pseudo-random number generation (extended abstract). In G.T. Blakley and D. Chaum, editors, *Advances in Cryptology — Proceedings of CRYPTO'84, Lecture Notes in Computer Science 196*, pages 193–202. Springer-Verlag, 1985.
- [298] E. R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. In B. Pfitzmann, editor, *Advances in Cryptology — Proceedings of EUROCRYPT'01, Lecture Notes in Computer Science 2045*, pages 195–210. Springer-Verlag, 2001.
- [299] D. Wheeler. Transactions using bets. In M. Lomas, editor, *Security Protocols, Lecture Notes in Computer Science 1189*, pages 89–92. Springer-Verlag, 1996.
- [300] M. Wiener. Cryptanalysis of short RSA secret exponents. *IEEE Transactions on Information Theory*, 36(3):553–558, 1990.
- [301] M. Wiener. Efficient DES key search. Technical report, TR-244, School of Computer Science, Carleton University, Ottawa, May 1994.
- [302] C.P. Williams and S.H. Clearwater. *Ultimate Zero and One*. Copernicus, Springer-Verlag New York, Inc., 2000.
- [303] T.Y.C. Woo and S.S. Lam. Authentication for distributed systems. *Computer*, 25(1):39–52, January 1992.
- [304] T.Y.C. Woo and S.S. Lam. A lesson on authentication protocol design. *Operating Systems Review*, 28(3):24–37, July 1994.
- [305] A.C. Yao. Theory and applications of trapdoor functions (extended abstract). In *Proceedings of 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 80–91, 1982.
- [306] T. Ylonen. The SSH (secure shell) remote login protocol. INTERNET-DRAFT, draft-ylonen-ssh-protocol-00.txt, September 1995.
- [307] T. Ylonen. SSH authentication protocol. INTERNET-DRAFT, draft-ietf-userauth-16.txt, September 2002.
- [308] T. Ylonen. SSH connection protocol. INTERNET-DRAFT, draft-ietf-connect-16.txt, September 2002.
- [309] T. Ylonen. SSH protocol architecture. INTERNET-DRAFT, draft-ietf-architecture-13.txt, September 2002.
- [310] T. Ylonen. SSH transport layer protocol. INTERNET-DRAFT, draft-ietf-transport-15.txt, September 2002.

- 
- [311] Y. Zheng. Digital signcryption or how to achieve  $\text{cost}(\text{signature} \& \text{encryption}) = \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$ . In B.S. Kaliski Jr., editor, *Advances in Cryptology — Proceedings of CRYPTO'97, Lecture Notes in Computer Science 1294*, pages 165–179. Springer-Verlag, 1997.
  - [312] Y. Zheng and J. Seberry. Immunizing public key cryptosystems against chosen ciphertext attacks. *Special Issue on Secure Communications, IEEE Journal on Selected Areas on Communications*, 11(5):715–724, June 1993.
  - [313] Y. Zheng and J. Seberry. Practical approaches to attaining security against adaptively chosen ciphertext attacks (extended abstract). In E.F. Brickell, editor, *Advances in Cryptology — Proceedings of CRYPTO'92, Lecture Notes in Computer Science 740*, pages 291–304. Springer-Verlag, 1993.
  - [314] P.R. Zimmermann. *The Official PGP User's Guide*. MIT Press, Cambridge, Massachusetts, 1995. Second printing.



[ G e n e r a l   I n f o r m a t i o n ]

书名 = 现代密码学理论与实践

作者 = ( 英 ) W e n b o   M a o 著      王继林      伍前红等译

页数 = 4 7 7

S S 号 = 1 1 3 0 9 0 4 2

出版日期 = 2 0 0 4 年 0 7 月

前言

第一部	分引言
第 1 章	一个简单的通信游戏
1 . 1	一个通信游戏
1 . 1 . 1	我们给出密码学的第一个应用示例
1 . 1 . 2	对密码学基础的初步提示
1 . 1 . 3	信息安全基础：计算困难性的背后
1 . 1 . 4	密码学的新作用：保证游戏的公平性
1 . 2	描述密码系统和协议的准则
1 . 2 . 1	保护的程度与应用需求相符合
1 . 2 . 2	对安全性的信心要依据所建立的“种系”
1 . 2 . 3	实际效率
1 . 2 . 4	采用实际的和可用的原型和服务
1 . 2 . 5	明确性
1 . 2 . 6	开放性
1 . 3	本章小结
习题	
第 2 章	防守与攻击
2 . 1	引言
2 . 1 . 1	本章概述
2 . 2	加密
2 . 3	易受攻击的环境（D o l e v - Y a o 威胁模型）
2 . 4	认证服务器
2 . 5	认证密钥建立的安全特性
2 . 6	利用加密的认证密钥建立协议
2 . 6 . 1	消息保密协议
2 . 6 . 2	攻击、修复、攻击、修复
2 . 6 . 3	消息认证协议
2 . 6 . 4	询问 - 应答协议
2 . 6 . 5	实体认证协议
2 . 6 . 6	一个使用公钥密码体制的协议
2 . 7	本章小结
习题	
第二部分	数学基础
标准符号	
第 3 章	概率论和信息论
3 . 1	引言
3 . 1 . 1	本章纲要
3 . 2	概率论的基本概念
3 . 3	性质
3 . 4	基本运算
3 . 4 . 1	加法规则
3 . 4 . 2	乘法规则
3 . 4 . 3	全概率定律
3 . 5	随机变量及其概率分布
3 . 5 . 1	均匀分布
3 . 5 . 2	二项式分布
3 . 5 . 3	大数定律
3 . 6	生日悖论
3 . 6 . 1	生日悖论的应用：指数计算的 P o l l a r d 袋鼠算法
3 . 7	信息论
3 . 7 . 1	熵的性质
3 . 8	自然语言的冗余度
3 . 9	本章小结
习题	
第 4 章	计算复杂性
4 . 1	引言
4 . 1 . 1	本章概述

- 4 . 2 图灵机
- 4 . 3 确定性多项式时间
- 4 . 3 . 1 多项式时间计算性问题
- 4 . 3 . 2 算法与计算复杂度表示
- 4 . 4 概率多项式时间
- 4 . 4 . 1 差错概率的特征
- 4 . 4 . 2 “总是快速且正确的”子类
- 4 . 4 . 3 “总是快速且很可能正确的”子类
- 4 . 4 . 4 “很可能快且总是正确的”子类
- 4 . 4 . 5 “很可能快且很可能正确的”子类
- 4 . 4 . 6 有效算法
- 4 . 5 非确定多项式时间
- 4 . 5 . 1 非确定多项式时间完全
- 4 . 6 非多项式界
- 4 . 7 多项式时间不可区分性
- 4 . 8 计算复杂性理论与现代密码学
- 4 . 8 . 1 必要条件
- 4 . 8 . 2 非充分条件
- 4 . 9 本章小结

#### 习题

### 第5章 代数学基础

- 5 . 1 引言
- 5 . 1 . 1 章节纲要
- 5 . 2 群
- 5 . 2 . 1 拉格朗日定理
- 5 . 2 . 2 群元素的阶
- 5 . 2 . 3 循环群
- 5 . 2 . 4 乘法群??
- 5 . 3 环和域
- 5 . 4 有限域的结构
- 5 . 4 . 1 含有素数个元素的有限域
- 5 . 4 . 2 模不可约多项式的有限域
- 5 . 4 . 3 用多项式基构造有限域
- 5 . 4 . 4 本原根
- 5 . 5 用椭圆曲线上的点构造群
- 5 . 5 . 1 群运算
- 5 . 5 . 2 点乘
- 5 . 5 . 3 椭圆曲线离散对数问题
- 5 . 6 本章小结

#### 习题

### 第6章 数论

- 6 . 1 引言
- 6 . 1 . 1 本章概述
- 6 . 2 同余和剩余类
- 6 . 2 . 1 ? n 中运算的同余性质
- 6 . 2 . 2 求解? n 中的线性同余式
- 6 . 2 . 3 中国剩余定理
- 6 . 3 欧拉 函数
- 6 . 4 费马定理、欧拉定理、拉格朗日定理
- 6 . 5 二次剩余
- 6 . 5 . 1 二次剩余的判定
- 6 . 5 . 2 勒让德 - 雅可比符号
- 6 . 6 模一个整数的平方根
- 6 . 6 . 1 求模为素数时的平方根
- 6 . 6 . 2 求模为合数时的平方根
- 6 . 7 B l u m 整数
- 6 . 8 本章小结

#### 习题

第三部分	基本的密码学技术
第 7 章	加密——对称技术
7 . 1	引言
7 . 1 . 1	本章概述
7 . 2	定义
7 . 3	代换密码
7 . 3 . 1	简单的代换密码
7 . 3 . 2	多表密码
7 . 3 . 3	弗纳姆密码和一次一密
7 . 4	换位密码
7 . 5	古典密码：使用和安全性
7 . 5 . 1	古典密码的使用
7 . 5 . 2	古典密码的安全性
7 . 6	数据加密标准（D E S）
7 . 6 . 1	介绍 D E S
7 . 6 . 2	D E S 的核心作用：消息的随机非线性分布
7 . 6 . 3	D E S 的安全性
7 . 7	高级加密标准（A E S）
7 . 7 . 1	R i j n d a e l 密码概述
7 . 7 . 2	R i j n d a e l 密码的内部函数
7 . 7 . 3	R i j n d a e l 内部函数的功能小结
7 . 7 . 4	快速而安全的实现
7 . 7 . 5	A E S 对应用密码学的积极影响
7 . 8	运行的保密模式
7 . 8 . 1	电码本模式（E C B）
7 . 8 . 2	密码分组链接模式（C B C）
7 . 8 . 3	密码反馈模式（C F B）
7 . 8 . 4	输出反馈模式（O F B）
7 . 8 . 5	计数器模式（C T R）
7 . 9	对称密码体制的密钥信道建立
7 . 1 0	本章小结
习题	
第 8 章	加密——非对称技术
8 . 1	引言
8 . 1 . 1	本章概述
8 . 2	“教科书式加密算法”的不安全性
8 . 3	D i f f i e - H e l l m a n 密钥交换协议
8 . 3 . 1	中间人攻击
8 . 4	D i f f i e - H e l l m a n 问题和离散对数问题
8 . 4 . 1	任意参数对于满足困难假设的重要性
8 . 5	R S A 密码体制（教科书式）
8 . 6	公钥密码体制的分析
8 . 7	R S A 问题
8 . 8	整数分解问题
8 . 9	教科书式 R S A 加密的不安全性
8 . 9 . 1	中间相遇攻击和教科书式 R S A 上的主动攻击
8 . 1 0	R a b i n 加密体制（教科书式）
8 . 1 1	教科书式 R a b i n 加密的不安全性
8 . 1 2	E l G a m a l 密码体制（教科书式）
8 . 1 3	教科书式 E l G a m a l 加密的不安全性
8 . 1 3 . 1	教科书式 E l G a m a l 加密的中间相遇攻击和主动攻击
8 . 1 4	公钥密码系统需要更强的安全定义
8 . 1 5	非对称密码与对称密码的组合
8 . 1 6	公钥密码系统密钥信道的建立
8 . 1 7	本章小结
习题	
第 9 章	理想情况下基本公钥密码函数的比特安全性
9 . 1	前言

- 9 . 1 . 1      本章概述
- 9 . 2        R S A 比特
- 9 . 3        R a b i n 比特
- 9 . 3 . 1      B l u m - B l u m - S h u b 伪随机比特生成器
- 9 . 4        E l G a m a l 比特
- 9 . 5        离散对数比特
- 9 . 6        本章小结

#### 习题

### 第 1 0 章      数据完整性技术

- 1 0 . 1      引言
- 1 0 . 1 . 1      本章概述
- 1 0 . 2      定义
- 1 0 . 3      对称技术
- 1 0 . 3 . 1      密码杂凑函数
- 1 0 . 3 . 2      基于密钥杂凑函数的 M A C
- 1 0 . 3 . 3      基于分组加密算法的 M A C
- 1 0 . 4      非对称技术    : 数字签名
- 1 0 . 4 . 1      数字签名的教科书式安全概念
- 1 0 . 4 . 2      R S A 签字体制 ( 教科书式版本 )
- 1 0 . 4 . 3      R S A 签字安全性的非形式化论证
- 1 0 . 4 . 4      R a b i n 签名体制 ( 教科书式版本 )
- 1 0 . 4 . 5      关于 R a b i n 签名的一个自相矛盾的安全性基础
- 1 0 . 4 . 6      E l G a m a l 签名体制
- 1 0 . 4 . 7      E l G a m a l 签名体制安全性的非形式化论证
- 1 0 . 4 . 8      E l G a m a l 签名族中的签名体制
- 1 0 . 4 . 9      数字签名体制安全性的形式化证明
- 1 0 . 5      非对称技术    : 无源识别的数据完整性
- 1 0 . 6      本章小结

#### 习题

### 第四部      分认证

### 第 1 1 章      认证协议——原理篇

- 1 1 . 1      引言
- 1 1 . 1 . 1      章节概述
- 1 1 . 2      认证和细化的概念
- 1 1 . 2 . 1      数据源认证
- 1 1 . 2 . 2      实体认证
- 1 1 . 2 . 3      认证的密钥建立
- 1 1 . 2 . 4      对认证协议的攻击
- 1 1 . 3      约定
- 1 1 . 4      基本认证技术
- 1 1 . 4 . 1      消息新鲜性和主体活现性
- 1 1 . 4 . 2      双方认证
- 1 1 . 4 . 3      包含可信第三方的认证
- 1 1 . 5      基于口令的认证
- 1 1 . 5 . 1      N e e d h a m 口令认证协议及其在 U N I X 操作系统中的实现
- 1 1 . 5 . 2      一次性口令机制 ( 及缺陷的修补 )
- 1 1 . 5 . 3      加盐操作 : 加密的密钥交换 ( E K E )
- 1 1 . 6      基于非对称密码学的认证密钥交换
- 1 1 . 6 . 1      工作站 - 工作站协议
- 1 1 . 6 . 2      简化 S T S 协议的一个缺陷
- 1 1 . 6 . 3      S T S 协议的一个瑕疵
- 1 1 . 7      对认证协议的典型攻击
- 1 1 . 7 . 1      消息重放攻击
- 1 1 . 7 . 2      中间人攻击
- 1 1 . 7 . 3      平行会话攻击
- 1 1 . 7 . 4      反射攻击
- 1 1 . 7 . 5      交错攻击
- 1 1 . 7 . 6      归因于类型缺陷攻击

1 1 . 7 . 7	归因于姓名遗漏攻击
1 1 . 7 . 8	密码服务滥用攻击
1 1 . 8	文献简记
1 1 . 9	本章小结
习题	
第 1 2 章	认证协议——实践篇
1 2 . 1	引言
1 2 . 1 . 1	章节概述
1 2 . 2	用于因特网的认证协议
1 2 . 2 . 1	I P 层通信
1 2 . 2 . 2	I P 安全协议 ( I P S e c )
1 2 . 2 . 3	因特网密钥交换 ( I K E ) 协议
1 2 . 2 . 4	I K E 中看似合理的可否认性
1 2 . 2 . 5	对 I P S e c 和 I K E 的批评意见
1 2 . 3	安全壳 ( S S H ) 远程登录协议
1 2 . 3 . 1	S S H 架构
1 2 . 3 . 2	S S H 传输层协议
1 2 . 3 . 3	S S H 策略
1 2 . 3 . 4	警告
1 2 . 4	K e r b e r o s 协议及其在 W i n d o w s 2 0 0 0 系统中的实现
1 2 . 4 . 1	单点登录结构
1 2 . 4 . 2	K e r b e r o s 交换
1 2 . 4 . 3	警告
1 2 . 5	S S L 和 T L S
1 2 . 5 . 1	T L S 架构概述
1 2 . 5 . 2	T L S 握手协议
1 2 . 5 . 3	T L S 握手协议的典型运行
1 2 . 5 . 4	对 T L S 协议的边信道攻击
1 2 . 6	本章小结
习题	
第 1 3 章	公钥密码的认证框架
1 3 . 1	前言
1 3 . 1 . 1	本章概述
1 3 . 2	基于目录的认证框架
1 3 . 2 . 1	证书发行
1 3 . 2 . 2	证书吊销
1 3 . 2 . 3	公钥认证框架实例
1 3 . 2 . 4	与 X . 5 0 9 公钥证书基础设施相关的协议
1 3 . 3	基于非目录的公钥认证框架
1 3 . 3 . 1	S h a m i r 的基于 I D 的签名方案
1 3 . 3 . 2	基于 I D 的密码确切提供了什么
1 3 . 3 . 3	自证实公钥
1 3 . 3 . 4	利用 “ 弱 ” 椭圆曲线对构造基于身份的公钥密码体制
1 3 . 3 . 5	S a k a i 、 O h g i s h i 和 K a s a h a r a 的基于 I D 的非交互密钥分享系
1 3 . 3 . 6	三方 D i f f i e - H e l l m a n 密钥协商
1 3 . 3 . 7	B o n e h 和 F r a n k l i n 的基于 I D 的密码体制
1 3 . 3 . 8	非交互特性：无密钥信道的认证
1 3 . 3 . 9	基于身份的公钥密码学的两个公开问题
1 3 . 4	本章小结
习题	
第五部分	建立安全性的形式化方法
第 1 4 章	公钥密码体制的形式化强安全性定义
1 4 . 1	引言
1 4 . 1 . 1	本章概述
1 4 . 2	安全性的形式化处理
1 4 . 3	语义安全性——可证明安全性的首次亮相
1 4 . 3 . 1	S R A 智力扑克协议



1 4 . 3 . 2	基于教科书式安全的安全性分析
1 4 . 3 . 3	G o l d w a s s e r 和 M i c a l i 的概率加密
1 4 . 3 . 4	G M 密码体制的安全性
1 4 . 3 . 5	E l G a m a l 体制的一种语义安全版本
1 4 . 3 . 6	基于 R a b i n 比特的语义安全密码体制
1 4 . 4	语义安全性的不充分性
1 4 . 5	超越语义安全性
1 4 . 5 . 1	抗击选择密文攻击的安全性
1 4 . 5 . 2	抗击适应性选择密文攻击的安全性
1 4 . 5 . 3	不可展密码学
1 4 . 5 . 4	不可区分性与不可展性的关系
1 4 . 6	本章小结
习题	
第 1 5 章	可证明安全的有效公钥密码体制
1 5 . 1	引言
1 5 . 1 . 1	本章概述
1 5 . 2	最优非对称加密填充
1 5 . 2 . 1	安全性证明的随机预言机模型
1 5 . 2 . 2	R S A - O A E P
1 5 . 2 . 3	R S A - O A E P 证明中的曲折
1 5 . 2 . 4	对 R S A - O A E P 的补救工作
1 5 . 2 . 5	R S A - O A E P “ 归约为矛盾 ” 的严谨性
1 5 . 2 . 6	对随机预言机模型的批评
1 5 . 2 . 7	作者对随机预言机模型价值的观点
1 5 . 3	C r a m e r - S h o u p 公钥密码体制
1 5 . 3 . 1	在标准困难性假设下的可证明安全性
1 5 . 3 . 2	C r a m e r - S h o u p 体制
1 5 . 3 . 3	安全性证明
1 5 . 4	可证明安全的混合密码体制综述
1 5 . 5	可证明安全的实用公钥密码体制的文献注记
1 5 . 6	本章小结
习题	
第 1 6 章	强可证明安全的数字签名方案
1 6 . 1	引言
1 6 . 1 . 1	本章纲要
1 6 . 2	数字签名的强安全性定义
1 6 . 3	E l G a m a l 族签名的强可证明安全
1 6 . 3 . 1	三元组 E l G a m a l 族签名
1 6 . 3 . 2	分叉归约技术
1 6 . 3 . 3	重行归约方法
1 6 . 4	适于应用的 R S A 和 R a b i n 签名方法
1 6 . 4 . 1	具有随机化填充的签名
1 6 . 4 . 2	概率签名方案
1 6 . 4 . 3	P S S - R : 消息可恢复的签名
1 6 . 4 . 4	签名和加密通用的 P S S - R 填充
1 6 . 5	签密
1 6 . 5 . 1	Z h e n g 的签密方案
1 6 . 5 . 2	一箭双雕 : 采用 R S A 签密
1 6 . 6	本章小结
习题	
第 1 7 章	分析认证协议的形式化方法
1 7 . 1	引言
1 7 . 1 . 1	本章概述
1 7 . 2	认证协议的形式化描述
1 7 . 2 . 1	加解密认证方法的不精确性
1 7 . 2 . 2	认证协议的细化描述
1 7 . 2 . 3	认证协议细化描述的例子
1 7 . 3	正确协议的计算观点——B e l l a r e - R o g a w a y 模型

1 7 . 3 . 1	参与者行为的形式模型化
1 7 . 3 . 2	相互认证的目标：匹配对话
1 7 . 3 . 3	M A P 1 协议及其安全性证明
1 7 . 3 . 4	协议正确性计算模型的进一步研究
1 7 . 3 . 5	讨论
1 7 . 4	正确协议的符号操作观点
1 7 . 4 . 1	定理证明
1 7 . 4 . 2	一种认证逻辑
1 7 . 5	形式化分析技术：状态系统探查
1 7 . 5 . 1	模型检验
1 7 . 5 . 2	N R L 协议分析机
1 7 . 5 . 3	C S P 方法
1 7 . 6	调和安全性形式化技术的两种观点
1 7 . 7	本章小结

## 习题

## 第六部分 密码学协议

## 第 1 8 章 零知识协议

1 8 . 1	引言
1 8 . 1 . 1	本章纲要
1 8 . 2	基本定义
1 8 . 2 . 1	计算模型
1 8 . 2 . 2	交互式证明协议的形式化定义
1 8 . 2 . 3	一个复杂性理论结果
1 8 . 3	零知识特性
1 8 . 3 . 1	完备零知识
1 8 . 3 . 2	诚实验证者的零知识
1 8 . 3 . 3	计算零知识
1 8 . 3 . 4	统计零知识
1 8 . 4	证明还是论据
1 8 . 4 . 1	零知识论据
1 8 . 4 . 2	零知识证明
1 8 . 5	双边差错协议
1 8 . 5 . 1	零知识证明双素整数
1 8 . 6	轮效率
1 8 . 6 . 1	子群成员归属的轮效率下界
1 8 . 6 . 2	离散对数的常数轮证明
1 8 . 7	非交互式零知识
1 8 . 7 . 1	利用指定验证者获得 N I Z K
1 8 . 8	本章小结

## 习题

## 第 1 9 章 回到“电话掷币”协议

1 9 . 1	B l u m “电话掷币”协议
1 9 . 2	安全性分析
1 9 . 3	效率
1 9 . 4	本章小结

## 第 2 0 章 结束语

## 参考文献